

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Sistemas

Análise de Séries Temporais e Modelagem Baseada em Regras Nebulosas

Tese de Doutorado submetida à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Energia Elétrica.

Por: Ivette R. Luna Huamaní

Orientador: Prof. Dr. Secundino Soares Filho

Co-orientadora: Profa. Dra. Rosangela Ballini

Banca Examinadora:

Prof. Dr. Aluizio Fausto Ribeiro Araújo - CIN/UFPE

Prof. Dr. Fernando José Von Zuben - FEEC/UNICAMP

Prof. Dr. Marinho Gomes de Andrade Filho - ICMC/USP - São Carlos

Prof. Dr. Paulo Sérgio Franco Barbosa - FEC/UNICAMP

Prof. Dr. Secundino Soares Filho (orientador)- FEEC/UNICAMP

Prof. Dr. Takaaki Ohishi - FEEC/UNICAMP

Campinas, SP

5 de outubro de 2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Luna Huamaní, Ivette R.

L971a Análise de séries temporais e modelagem baseada em
regras nebulosas / Ivette R. Luna Huamaní. –Campinas, SP:
[s.n.], 2007.

Orientador: Secundino Soares Filho.

Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas especialistas (Computação). 2. Redes
neurais (Computação). 3. Inteligência artificial. 4. Sistemas
difusos. 5. Aprendizado do computador. 6. Análise de
series temporais - Processamento de dados. I. Soares Filho,
Secundino. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título

Título em Inglês:	Time series analysis and modeling based on fuzzy rules the School of Electrical and Computer Engineering
Palavras-chave em Inglês:	Expert systems, Neural networks, Artificial intelligence, Fuzzy systems, Learning algorithms, Fuzzy logic Intelligent agents (software), Distributed operating systems
Área de concentração:	Energia Elétrica
Titulação:	Doutor em Engenharia Elétrica
Banca Examinadora:	Aluizio Fausto Ribeiro Araújo, Fernando José Von Zuben, Marinho Gomes de Andrade Filho, Paulo Sérgio Franco Barbosa, Takaaki Ohishi
Data da defesa:	05/10/2007
Programa de Pós-Graduação:	Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidata: Ivette Raymunda Luna Huamaní

Data da Defesa: 5 de outubro de 2007

Título da Tese: "Análise de Séries Temporais e Modelagem Baseada em Regras Nebulosas"

Prof. Dr. Secundino Soares Filho (Presidente):

Secundino Soares F.

Prof. Dr. Aluizio Fausto Ribeiro Araujo:

Aluizio F. Ribeiro Araujo

Prof. Dr. Marinho Gomes de Andrade Filho:

Marinho Gomes de Andrade Filho

Prof. Dr. Paulo Sérgio Franco Barbosa:

Dr. PSFB

Prof. Dr. Fernando José Von Zuben:

Fernando José Von Zuben

Prof. Dr. Takaaki Ohishi:

Takaaki Ohishi

Resumo

Este trabalho propõe uma metodologia baseada em regras nebulosas para a modelagem e previsão de séries temporais. Inicialmente, os dados são pré-processados para, a seguir, ocorrer a seleção de variáveis que serão utilizadas pelos modelos de série temporal. Para essa finalidade, nesta tese propõe-se um conjunto de aproximações necessárias para o cálculo do critério de informação mútua parcial, o qual é a base para o algoritmo de seleção de entradas utilizado. A próxima etapa corresponde à determinação da estrutura do modelo e ajuste dos parâmetros. Com o intuito de definir de forma automática a estrutura do modelo, de forma simultânea ao ajuste dos parâmetros, dois algoritmos de aprendizado construtivo - *offline* e *online* são propostos. Ambos os algoritmos utilizam como base para o seu desenvolvimento o algoritmo da maximização da verossimilhança, assim como critérios de geração e punição (ou poda) de regras nebulosas. Finalmente, o modelo obtido é validado e aplicado na previsão de um e vários passos à frente. Análises comparativas são apresentadas utilizando séries temporais sintéticas e de problemas reais. Os resultados mostram que as propostas deste trabalho são uma alternativa eficiente para a modelagem e previsão de séries temporais.

Palavras-chave: Análise de séries temporais, modelos baseados em regras nebulosas, algoritmo EM, aprendizado construtivo, aprendizado seqüencial.

Abstract

This work presents a methodology for time series modeling and forecasting. First, the methodology considers the data pre-processing and the system identification, which implies on the selection of a suitable set of input variables for modeling the time series. In order to achieve this task, this work proposes an algorithm for input selection and a set of approximations that are necessary for estimating the partial mutual information criterion, which is the base of the algorithm used at this stage. Then, the model is built and adjusted. With the aim of performing an automatic structure selection and parameters adjustment simultaneously, this thesis proposes two constructive learning algorithms, namely offline and online. These algorithms are based on the Expectation Maximization optimization technique, as well as on adding and pruning operators of fuzzy rules that are also proposed in this work. Finally, models are validated and applied to one-step ahead and multi-step ahead forecasting. Comparative analysis using synthetic and real time series are detailed. The results show the adequate performance of the proposed approach and presents it as a promising alternative for time series modeling and forecasting.

Keywords: Time series analysis, fuzzy rules based models, EM algorithm, constructive learning, sequential learning.

Agradecimentos

À professora Rosangela Ballini pelo apoio e orientação fundamentais para o desenvolvimento deste trabalho.

Ao professor Secundino Soares, pelo voto de confiança e pela oportunidade de trabalharmos no mesmo grupo de pesquisa.

À Fundação de Amparo à Pesquisa do estado de São Paulo - FAPESP, pelo apoio financeiro.

À minha família, e principalmente ao meu Ed, pelo carinho, paciência e força oferecidos em todo momento e sem condições.

Ao meu filho, Matheus, pelos momentos de alegria concedidos e por ser a minha fonte inesgotável de força e energia.

Aos meus amigos do laboratório pelo apoio e pela amizade incondicional oferecida.

Finalmente, a Deus, por ter me dado a benção da vida, assim como os melhores presentes: a minha família e os meus amigos.

Para o Ed e para o Ma :)

Sumário

Lista de figuras

Lista de Figuras

Lista de Tabelas

Lista de Abreviações p. 19

Lista de Símbolos p. 21

1 Introdução p. 25

1.1 Enfoque da pesquisa e motivação para o desenvolvimento deste trabalho . . p. 25

1.2 Objetivos p. 29

1.3 Organização do trabalho p. 30

2 Pré-processamento e seleção de entradas p. 33

2.1 Introdução p. 33

2.2 Pré-processamento dos dados p. 35

2.3 Critério de informação de Bayes - BIC p. 36

2.4 Técnicas não paramétricas de seleção de entradas p. 37

2.4.1 Algoritmo dos falsos vizinhos mais próximos - FNN p. 38

2.4.1.1 Algoritmo FNN p. 40

2.4.2 Critério de informação mútua parcial - PMI p. 41

2.4.2.1	Aproximação das funções de probabilidades marginal e conjunta	p. 42
2.4.2.2	Aproximação do valor esperado condicional	p. 48
2.4.2.3	Determinação do limiar de confiança para o PMI	p. 48
2.4.2.4	Algoritmo PMI	p. 49
2.4.3	Abordagem combinada: algoritmo FNN-PMI	p. 50
2.5	Resumo	p. 50
3	Modelos não lineares de séries temporais baseados em regras nebulosas	p. 51
3.1	Revisão bibliográfica	p. 51
3.2	Modelos comparativos	p. 56
3.3	Resumo	p. 58
4	Estrutura de modelo de série temporal	p. 59
4.1	Introdução	p. 59
4.2	Estrutura geral do modelo	p. 59
4.3	Resumo	p. 64
5	Modelo baseado em regras nebulosas com aprendizado construtivo <i>offline</i>	p. 65
5.1	Introdução	p. 65
5.2	Modelo construtivo baseado em regras nebulosas (C-FSM)	p. 66
5.2.1	Primeira etapa: inicialização	p. 66
5.2.1.1	O algoritmo EM	p. 66
5.2.2	Segunda etapa: treinamento e adaptação	p. 69
5.2.2.1	Geração de uma nova regra o adição	p. 70
5.2.2.2	Eliminação ou poda	p. 72

5.3	Um exemplo demonstrativo	p. 74
5.4	Resumo	p. 78
6	Modelo baseado em regras nebulosas com aprendizado <i>online</i>	p. 79
6.1	Introdução	p. 79
6.2	Modelo baseado em regras nebulosas com aprendizado <i>online</i> ou seqüencial	p. 80
6.2.1	Primeira etapa: inicialização	p. 81
6.2.1.1	O algoritmo de agrupamento subtrativo - SC	p. 82
6.2.2	Segunda etapa: adaptação	p. 84
6.2.2.1	Geração de novas regras	p. 86
6.2.2.2	Eliminação ou poda	p. 87
6.3	Exemplo demonstrativo	p. 88
6.4	Resumo	p. 91
7	Modelagem e previsão de séries temporais	p. 93
7.1	Introdução	p. 93
7.2	Série de fornalha de gás Box-Jenkins	p. 93
7.3	Série econômica	p. 97
7.4	Séries financeiras	p. 103
7.4.1	Série NN3_102	p. 104
7.4.2	Série NN3_104	p. 108
7.4.3	Série NN3_109	p. 111
7.5	Série de vazões mensais	p. 115
7.6	Série de vazões semanais	p. 130
7.7	Resumo	p. 134
8	Conclusões e trabalhos futuros	p. 137

8.1	Conclusões	p. 137
-----	----------------------	--------

8.2	Trabalhos futuros	p. 139
-----	-----------------------------	--------

Apêndice A – Trabalhos publicados pela autora	p. 141
--	--------

Referências	p. 143
--------------------	--------

Lista de Figuras

1	Exemplo de um falso vizinho.	p. 40
2	Aproximação de distribuições de probabilidade utilizando diferentes funções de <i>kernel</i>	p. 44
3	Exemplo de dependência não-linear entre duas variáveis.	p. 46
4	Estrutura do modelo proposto composto por M regras nebulosas.	p. 60
5	Condição para gerar uma nova regra nebulosa.	p. 71
6	Série de fornalha de gás Box-Jenkins.	p. 74
7	Localização dos centros e curvas de nível associadas para diferentes estágios do processo de aprendizado construtivo <i>offline</i> , para o modelo da fornalha de gás Box-Jenkins.	p. 76
8	Ativação das regras nebulosas sob o conjunto de treinamento.	p. 77
9	Localização dos centros e curvas de nível associadas para diferentes estágios do processo de aprendizado seqüencial, para o modelo da fornalha de gás Box-Jenkins.	p. 90
10	Variação de M durante o aprendizado dos modelos baseados em regras para a série de fornalha de gás Box-Jenkins: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 95
11	Previsão um passo à frente da série de fornalha de gás Box-Jenkins: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 96
12	Análise dos resíduos para o modelo C-FSM : (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.	p. 100
13	Análise dos resíduos para o modelo C-FSM_t : (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.	p. 100

14	Análise dos resíduos para o modelo A-FSM : (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.	p. 101
15	Análise dos resíduos para o modelo mA-FSM : (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.	p. 101
16	Previsão até 5 passos à frente dos modelos baseados em regras nebulosas da taxa de desemprego trimestral dos Estados Unidos.	p. 103
17	Variação do ERMs de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_102.	p. 107
18	Previsão vários passos à frente para a série NN3_102: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 107
19	Variação do ERMs de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_104.	p. 110
20	Previsão vários passos à frente para a série NN3_104: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 111
21	Variação do ERMs de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_109.	p. 114
22	Previsão vários passos à frente para a série NN3_109: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 114
23	Definição do conjunto inicial de possíveis entradas via o algoritmo FNN, para a série mensal do posto de Furnas: (a) modelo C-FSM (65 anos), (b) modelo A-FSM (20 anos), (c) modelo C-FSM (50 anos).	p. 116
24	Análise dos resíduos para a série de Furnas utilizando o C-FSM na previsão um passo à frente, período de 1981 a 1985: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.	p. 120
25	Análise dos resíduos para a série de Furnas utilizando o C-FSM na previsão um passo à frente, período de 1996 a 2005: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.	p. 120
26	Análise dos resíduos para a série de Furnas utilizando o A-FSM na previsão um passo à frente: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.	p. 121

27	Análise dos resíduos para a série de Furnas utilizando o mA-FSM na previsão um passo à frente: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.	p. 121
28	Variação de M para cada um dos modelos baseados em regras, ajustados para a série de vazões mensais do posto de Furnas: (a) C-FSM (período de 1981 a 1985), (b) C-FSM (período de 1996 a 2005), (c) A-FSM , (d) mA-FSM	p. 122
29	Resultados de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1981 a 1985: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 123
30	Resultados de previsão um passo à frente para a série de vazões mensais do posto de Furnas: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 124
31	Definição do conjunto inicial de possíveis entradas via o algoritmo FNN, para a série semanal do posto de Furnas: (a) modelo C-FSM (68 anos), (b) modelo A-FSM (20 anos).	p. 131
32	Variação de M para cada um dos modelos baseados em regras ajustados para a série de vazões semanais do posto de Furnas: (a) C-FSM , (b) A-FSM , (c) mA-FSM	p. 132

Lista de Tabelas

1	Comparação do critério de informação mútua e do coeficiente de correlação.	p. 47
2	Erros de previsão um passo à frente para a série de fornalha de gás Box-Jenkins	p. 95
3	Desempenho dos modelos de previsão de um até cinco passos à frente para a série trimestral de desemprego dos Estados Unidos: erro quadrático médio (EQM).	p. 102
4	Resultados do algoritmo FNN-PMI para a série NN3_102.	p. 105
5	Erros de previsão para a série NN3_102.	p. 106
6	Resultados do algoritmo FNN-PMI para a série NN3_104.	p. 109
7	Erros de previsão para a série NN3_104.	p. 110
8	Resultados do algoritmo FNN-PMI para a série NN3_109.	p. 112
9	Erros de previsão para a série NN3_109.	p. 113
10	Resultados aplicando o algoritmo FNN-PMI para a série de vazões mensais do posto de Furnas.	p. 117
11	Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1981 a 1985.	p. 124
12	Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1996 a 2005.	p. 125
13	Erros mensais para os modelos C-FSM e A-FSM , período de 1996 a 2005.	p. 126
14	Seleção de entradas para os modelos construtivos, ordem dos modelos auto-regressivos e número de regras dos modelos RNN para a série de vazões mensais de Furnas.	p. 127
15	Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, modelagem mês a mês para o período de 1996 a 2005.	p. 128

16	Erros globais para os modelos C-FSM e A-FSM , modelagem mês a mês para o período de 1996 a 2005.	p. 129
17	Resultados aplicando o algoritmo FNN-PMI para a série de vazões semanais do posto de Furnas.	p. 132
18	Erros de previsão um passo à frente para a série de vazões semanais do posto de Furnas.	p. 133

Lista de Abreviações

FNN	- Falsos Vizinhos mais Próximos ou <i>False Nearest Neighbors</i>
PMI	- Informação Mútua Parcial ou <i>Partial Mutual Information</i>
MI	- Informação Mútua ou <i>Mutual Information</i>
BIC	- Critério de Informação Bayesiano ou <i>Bayesian Information Criterion</i>
C-FSM	- Modelo Construtivo Baseado em Regras Nebulosas ou <i>Constructive-Fuzzy System Model</i>
A-FSM	- Modelo Adaptativo Baseado em Regras Nebulosas ou <i>Adaptive-Fuzzy System Model</i>
mA-FSM	- Modelo Adaptativo Modificado Baseado em Regras Nebulosas ou <i>modified Adaptive-Fuzzy System Model</i>
EM	- algoritmo da maximização da verossimilhança ou <i>Expectation Maximization algorithm</i>
TS	- modelos do tipo <i>Takagi-Sugeno</i>
SC	- algoritmo SC ou <i>Subtractive Clustering</i>
wRLS	- algoritmo dos mínimos quadrados recursivo ponderado ou <i>weighted Recursive Least Square algorithm</i>
eTS	- modelo evolutivo baseado em regras nebulosas ou <i>evolving TS fuzzy model</i>
simpleTS	- modificação do modelo eTS
REQM	- raiz do erro quadrático médio
NDEI	- índice adimensional do erro
EQM	- erro quadrático médio

ERMs	- erro relativo médio simétrico
ERM	- erro relativo médio
EAM	- erro absoluto médio
ARIMA	- Modelo Autoregressivo Médias Móveis Integrado ou <i>linear univariate AutoRegressive Integrated Moving Average model</i>
MSA	- Modelos Autoregressivos baseados em regimes Markovianos ou <i>Markov Switching Autoregressive model</i>
TAR	- Modelos não lineares na média limiar autoregressivo ou <i>Threshold AutoRegressive model</i>
PAR	- Modelo Periódico Autoregressivo ou <i>Periodic AutoRegressive model</i>
ANFIS	- Sistema de Inferência Neural Nebuloso Adaptativo ou <i>Adaptive NeuroFuzzy Inference System</i>
RNN	- Rede Neural Nebulosa
MLP	- Rede Neural Multicamadas ou <i>MultiLayer Perceptron neural network</i>
PREVIVAZ	- Sistema de PREVisão de VAZões Naturais Semanais utilizado pelo setor elétrico brasileiro

Lista de Símbolos

γ -th	- percentil em $\gamma\%$
p	- ordem do modelo ou número de entradas
N	- número de dados
$\hat{\sigma}_x^2$	- variância estimada associada a variável x
ρ	- coeficiente de correlação
z_t	- série temporal estacionária
$f_{xy}(x_i, y_i)$	- função de distribuição de probabilidade conjunta das variáveis residuais x_i e y_i
$f_x(x_i)$	- função de probabilidade marginal de x , considerando a variável residual x_i
$K(\cdot)$	- função de Kernel
λ	- parâmetro de dispersão ou largura de banda para o cálculo das funções de probabilidade $f_{xy}(x_i, y_i)$ e $f_x(x_i)$
σ_x	- desvio padrão de x
$E(x_i \mathbf{z})$	- valor esperado condicional de x_i , dado \mathbf{z}
μ	- média histórica
$\mu(m)$	- média histórica mensal, para $m = 1, \dots, 12$
$std(\cdot)$	- desvio padrão
M	- número de regras nebulosas que compõem o modelo
M^0	- número inicial de regras nebulosas do sistema baseado em regras
k	- instante de tempo
\mathbf{x}^k	- vetor de entrada no instante k

- x_i^k - i -ésimo componente do vetor de entrada no instante k
- \mathbf{y}^k - vetor ou variável de saída desejada no instante k
- Z_0^+ - conjunto de números inteiros positivos
- M^k - número de regras nebulosas que compõem o modelo no instante k
- R_i^k - i -ésima regra nebulosa no instante k
- \mathcal{L} - função objetivo do algoritmo EM equivalente ao logaritmo da função de verossimilhança do modelo a ajustar
- D - conjunto de pares entrada-saída
- Ω - conjunto de todos os parâmetros do modelo
- \mathbf{C} - conjunto de parâmetros dos antecedentes das regras nebulosas
- y_i^k - saída estimada do i -ésimo modelo local no instante k
- g_i^k - grau de pertinência associada a i -ésima regra nebulosa no instante k
- \hat{y}^k - saída estimada do modelo no instante k
- \mathbf{V}_i - matriz de covariância associada a i -ésima regra nebulosa
- \mathbf{c}_i - centro associado a i -ésima regra nebulosa
- ϕ^k - vetor composto pelos componentes de entrada e um termo constante para a representação dos consequentes das regras nebulosas
- θ_i - coeficientes dos modelos lineares que definem os consequentes das regras nebulosas
- \mathbf{V}_i^k - matriz de covariância associada a i -ésima regra nebulosa no instante k
- \mathbf{c}_i^k - centro associado a i -ésima regra nebulosa no instante k
- θ_i^k - coeficientes dos modelos lineares que definem os consequentes das regras nebulosas no instante k
- h_i^k - estimativa a posteriori do grau de pertinência associado a i -ésima regra nebulosa no instante k

$P(i \mathbf{x}^k)$	- probabilidade condicional de i dado o vetor de entrada \mathbf{x} no instante k
$P(y^k \mathbf{x}^k, \theta_i)$	- probabilidade da saída desejada ser gerada pela i -ésima regra nebulosa, dada a entrada \mathbf{x}^k e os coeficientes θ_i no instante k
σ_i^2	- variância estimada para a i -ésima saída do modelo baseado em regras nebulosas
α_i	- índice de ativação da i -ésima regra nebulosa
α_i^k	- coeficiente positivo necessário para o cálculo das funções de pertinência, $i = 1, \dots, M$, no instante k
α_{min}	- limiar para os coeficientes α_i utilizado para a eliminação ou poda de regras nebulosas
N^0	- número de dados entrada-saída utilizados na inicialização dos modelos baseados em regras nebulosas
ψ^k	- vetor entrada-saída equivalente a $[\mathbf{x}^k \ y^k]$
P_k	- potencial calculado pelo algoritmo SC para o k -ésimo dado entrada-saída
P_1^*	- potencial máximo que determina o primeiro centro utilizando o algoritmo SC
r_a, r_b	- parâmetros utilizados no algoritmo SC
$\bar{\varepsilon}, \underline{\varepsilon}, d_{min}$	- coeficientes utilizados no algoritmo SC, nest trabalho considerados constantes
T	- janela de tempo utilizada nas aproximações sequenciais no ajuste do modelo A-FSM
\mathbf{C}_i^k	- matriz de covariância para a estimativa recursiva dos coeficientes θ_i^k do modelo A-FSM
f_{forget}^k	- fator de esquecimento utilizado no ajuste recursivo de θ_i^k para o modelo A-FSM
τ_i^k	- índice de permanência da i -ésima regra nebulosa no instante k
Δx^k	- equivalente a $x^k - x^{k-1}$
$\hat{\rho}_k$	- coeficiente de autocorrelação parcial estimada
\hat{s}_k	- coeficiente de correlação cruzada estimada

\hat{a}_k - resíduo estimado

n - número de parâmetros livres ou graus de liberdade do modelo

y^k - saída real no instante k

1 Introdução

1.1 Enfoque da pesquisa e motivação para o desenvolvimento deste trabalho

Modelos de séries temporais têm sido amplamente utilizados nas mais diversas formas, dependendo da área de aplicação, como filtragem, predição de valores futuros, simulação ou a descrição matemática da série temporal que, de alguma maneira, forneça algum tipo de informação adicional ou explicação referente ao problema em estudo (Brockwell e Davis, 2002).

Em termos gerais, o principal objetivo do estudo de séries temporais é a construção de modelos no domínio do tempo ou no domínio da frequência, com o intuito de investigar o mecanismo gerador da série temporal, assim como fazer previsões de valores futuros, a curto, médio e longo prazo, com o conhecimento de observações passadas.

Quando um modelo de série temporal é construído, em muitos casos considera-se apenas a existência de relações lineares entre as variáveis dependentes e independentes. Geralmente a modelagem é baseada em modelos lineares como é o caso dos modelos lineares auto-regressivos (AR), ou auto-regressivos médias móveis (ARMA) (Box et al., 1994), (Hamilton, 1994). A estratégia para a construção destes modelos é baseada em um ciclo iterativo, no qual a seleção da estrutura do modelo é baseada nos próprios dados (Ballini, 2000). De acordo com Morettin e Tolo (2004), os estágios do ciclo iterativo utilizado na construção de modelos lineares de séries temporais são:

1. A análise dos dados, onde verifica-se a natureza da série, as suas componentes (tendência, sazonalidade, etc) e se existe a necessidade de transformar ou não a série;
2. Uma classe de modelos é analisada (etapa de especificação);
3. Identificação do modelo, com base na análise de autocorrelação, autocorrelação parcial,

ou ainda, critérios tais como o critério de informação bayesiana (BIC);

4. A seguir, vem a fase de estimação, na qual os parâmetros do modelo identificado são estimados;
5. Finalmente, há a verificação do modelo ajustado por meio da análise dos resíduos, para averiguar se este é adequado para os fins em vista, no caso, para a previsão.

Caso o modelo não seja adequado, o procedimento acima é repetido, voltando à fase de identificação. Um procedimento que muitas vezes é utilizado é identificar não somente um único modelo, mas alguns modelos que serão estimados e verificados. Assim, a fase crítica do procedimento acima é a identificação do modelo, sendo possível que diferentes modelos sejam identificados para a mesma série temporal (Morettin e Toloi, 2004). Deve-se observar que a etapa de análise dos dados antecede a identificação e seleção do modelo, sendo a análise realizada fundamental para os estágios seguintes, já que a análise efetuada previamente servirá de base para a realização de uma escolha adequada de modelo de série temporal.

A utilização de modelos lineares de séries temporais, traz algumas vantagens e desvantagens. Como vantagens, destacam-se a facilidade de interpretação do modelo, assim como uma clara visualização das relações lineares existentes entre as variáveis dependentes e a variável dependente, pois os coeficientes medem de forma indireta essa relação. Outra vantagem importante é a facilidade de manipulação dos modelos, tanto na fase de seleção e ajuste, como na fase de validação, além do curto tempo de processamento para o ajuste dos coeficientes que definem o modelo. Por outro lado, estes não são capazes de lidar com relações não-lineares existentes entre as variáveis e, como é sabido, problemas reais envolvem na maioria das vezes relações não lineares.

O ciclo de passos mencionado e normalmente seguido na construção de modelos lineares de séries temporais pode ser adaptado para o ajuste de modelos não-lineares de séries temporais baseados na inteligência computacional. Nos últimos anos alguns trabalhos relevantes e direcionados neste assunto tem surgido na literatura, como é o caso do trabalho descrito por Tsay (2005). Neste livro, mais especificamente no Capítulo 4, o modelo linear é substituído por um modelo de rede neural multicamadas (Haykin, 2001b), sendo o modelo validado e analisado de forma similar aos modelos lineares ajustados para o mesmo problema, mostrando inclusive, o enfoque estatístico da rede neural, assim como as suas semelhanças com as abordagens estatísticas convencionais. As adaptações da sequência de passos descrita para o caso de modelos não-lineares baseados na inteligência computacional deve-se ao fato dos modelos não serem efetivamente lineares.

Diversas abordagens neurais e nebulosas vêm sendo fortemente utilizadas para fins de previsão, tendo como objetivo aproveitar as dependências lineares e não-lineares entre as variáveis de entrada e saída, como uma maneira de extrair a maior quantidade de informação possível a partir dessas variáveis e obter assim, modelos de séries temporais mais eficientes. Estes modelos vêm mostrando resultados bastante promissores quando utilizados na modelagem e previsão de séries temporais, sendo estes aplicados nas mais diversas áreas como economia, física, engenharia (Nie, 1997) e recursos hídricos (Zealand et al., 1999), (Luna et al., 2005). Assim, os modelos baseados na inteligência computacional tornam-se alternativas atraentes para o estudo das séries temporais, devido à sua capacidade de tratar as incertezas que podem estar presentes nos dados, a capacidade de realizar mapeamentos não-lineares, assim como a adaptação a possíveis mudanças na dinâmica da série quando se trata de modelos adaptativos.

Por outro lado, devido à própria natureza não-linear dos modelos baseados na inteligência computacional, surge a necessidade de desenvolver algoritmos de seleção de entrada compatíveis, de tal forma que o aumento na complexidade do modelo (de linear a não-linear) seja justificado pela obtenção de modelos adequados e eficientes, que possam explorar e aproveitar ao máximo as informações transmitidas, de forma independente ao tipo de relação existente entre as variáveis de entrada e de saída (linear e não-linear).

Além disso, embora o conhecimento e a experiência do especialista, que ajusta e define os modelos de previsão a utilizar seja de extrema importância, como mostrado em diferentes eventos mundiais de previsão de séries temporais (veja-se (?) e (?), por exemplo), estes mesmos eventos mostraram que, a automatização de algumas etapas do processo de obtenção de modelos de séries temporais, aceleram o processo de modelagem, tornado-se uma excelente ferramenta de suporte para o especialista, principalmente em problemas de previsão de grande porte. A adoção de uma metodologia se encaixa perfeitamente nesse sentido, pois não só define os passos a seguir, mas também formaliza o processo, permitindo a sua aplicação a conjuntos de séries temporais com características variadas.

Assim, este trabalho propõe uma metodologia para a construção de modelos de séries temporais baseados na inteligência computacional, tendo como ponto de partida os estágios que compõem o ciclo iterativo descrito para a geração de modelos lineares de séries temporais.

A metodologia proposta é composta por cinco estágios, os quais são delineados como segue:

1. *Análise dos dados e pré-processamento:* a série temporal é analisada e pré-processada, de acordo com a natureza da série, verificando a existência dos componentes de tendên-

cia e sazonalidade, determinando se necessário, a transformação mais adequada para os próximos estágios do processo.

2. Seleção das variáveis de entrada a utilizar: as variáveis necessárias para modelar a série temporal são identificadas. Para esta etapa, este trabalho propõe uma abordagem que combina o algoritmo dos Falsos Vizinhos mais Próximos (FNN), proposto por Abarbanel e Kennel (1993) e o critério de Informação Mútua Parcial (PMI), originalmente proposto por Sharma (2000). A vantagem de usar o algoritmo proposto é que tanto as dependências lineares e não-lineares entre as variáveis de entrada e saída são consideradas na modelagem da série, tendo assim um melhor aproveitamento das informações transmitidas ao modelo. Além disso as estimativas propostas para o cálculo do critério PMI aceleram o processo iterativo quando comparado com a proposta inicial apresentada em (Sharma, 2000).

3. Definição da estrutura do modelo e ajuste dos parâmetros: uma vez que o conjunto de variáveis de entrada a utilizar é definido, procede-se com a definição da estrutura do modelo. A inicialização da estrutura e dos parâmetros do modelo faz parte desta etapa. Os modelos propostos neste trabalho são baseados em regras nebulosas do tipo Takagi-Sugeno ou TS (Takagi e Sugeno, 1985), sendo a estrutura do modelo e os seus parâmetros ajustados simultaneamente aplicando dois algoritmos de aprendizado diferentes. O primeiro algoritmo é baseado em uma abordagem construtiva *offline*, resultando nos modelos denominados de modelos construtivos baseados em regras nebulosas (**C-FSM**). O segundo algoritmo é baseado em uma abordagem seqüencial, obtendo assim os modelos chamados de modelos adaptativos baseados em regras nebulosas (**A-FSM**) ou modelos adaptativos modificados baseados em regras nebulosas **mA-FSM**, o qual é uma modificação do modelo original **A-FSM**. Os algoritmos de aprendizado propostos surgem a partir do algoritmo da maximização da verossimilhança ou algoritmo EM (Jordan e Jacobs, 1993), o qual é amplamente utilizado para o ajuste de modelos de redes neurais. Desta forma, este trabalho estendeu a aplicação do algoritmo EM para a área de sistemas de inferência nebulosos, adaptando o algoritmo de acordo com as especificações dos modelos ajustados, sejam estes construtivos *offline* ou *online*. As principais vantagens dos modelos propostos são:
 - A definição automática da estrutura do modelo, realizado de forma simultânea ao ajuste dos parâmetros;
 - A interpretação da base de regras nebulosas codificada na estrutura do modelo;

- O aprendizado seqüencial proporciona modelos com capacidade de adaptação a variações na dinâmica da série, como é o caso por exemplo de série temporais com quebras no domínio do tempo.
 - Além disso, os modelos com aprendizado seqüencial requerem de um menor esforço computacional, já que, a cada iteração modelo processa apenas o dado recentemente conhecido. Já no caso do modelo com aprendizado construtivo *offline*, o total do conjunto de dados de treinamento são processados a cada etapa ou iteração do processo de otimização.
4. Adequação do modelo obtido: a adequação dos modelos obtidos no final do processo de otimização é verificada por meio de uma análise dos resíduos. No caso do modelo construtivo *offline* a adequação do modelo é verificada no final do ajuste do modelo. No caso dos modelos *online*, os resíduos e a respectiva análise dos mesmos é realizada de forma permanente ou enquanto o modelo é ajustado.
 5. Validação: finalmente, o modelo é validado verificando o desempenho dos mesmos na previsão um e múltiplos passos à frente, através de métricas de erro previamente definidas.

As inovações propostas neste trabalho serão descritas em maior detalhe nos respectivos capítulos que compõem esta tese. A seguir citam-se os objetivos deste trabalho.

1.2 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de uma metodologia para a análise, modelagem e previsão de séries temporais, envolvendo o estudo das diferentes etapas que compõem o processo. Esta metodologia inclui:

- A influência do número de dados disponíveis no processo de modelagem e as suas diferentes etapas;
- As transformações necessárias a serem aplicadas aos dados disponíveis, verificando a necessidade de cada uma das operações utilizadas, assim como a influência das transformações utilizadas no desempenho dos modelos obtidos;
- Extensões concernentes ao critério de seleção de entrada utilizado, desde a seleção de um conjunto inicial de possíveis entradas para a modelagem da série temporal, até a escolha das variáveis de entrada mais relevantes;

- A determinação de uma estrutura adequada para cada série temporal analisada;
- A aplicação da metodologia proposta em problemas da literatura, assim como em problemas reais. Os resultados obtidos utilizando a proposta deste trabalho são comparados com os resultados obtidos por outros modelos similares da literatura, tanto lineares como não-lineares.

Além disso, este trabalho fornece um conjunto de estudos de caso, os quais têm por objetivo:

- A avaliação dos diferentes modelos de séries temporais propostos e utilizados, assim como o desempenho destes para diferentes tipos de séries temporais, considerando as características e dinâmicas particulares de cada problema analisado;
- Um melhor entendimento das distintas abordagens utilizadas, lineares e não-lineares, assim como as vantagens e desvantagens quando adotadas.

1.3 Organização do trabalho

Este capítulo apresentou o contexto em que este trabalho foi desenvolvido e a motivação para o desenvolvimento dos modelos propostos.

O Capítulo 2 apresenta em detalhe os critérios FNN e PMI adotados para a seleção das variáveis de entrada a serem utilizadas na construção dos modelos de séries temporais, assim como uma metodologia para combiná-los.

Este capítulo apresenta também uma breve descrição do critério BIC (*Bayesian Information Criterion*) (Schwarz, 1978), o qual é empregado para a seleção de variáveis de entrada no modelo periódico autoregressivo (PAR), ajustado para uma série de vazões mensais.

O Capítulo 3 apresenta uma revisão bibliográfica sobre os diferentes modelos não-lineares baseados em regras nebulosas, atualmente empregados na literatura na previsão de séries temporais. Dentre eles, destacam-se os modelos adaptativos ou evolutivos, com ênfase nos modelos a serem utilizados para comparação de desempenho com os modelos propostos neste trabalho.

A estrutura geral do modelo baseado em regras nebulosas proposto neste trabalho é descrita no Capítulo 4. Para a determinação da estrutura e ajuste dos parâmetros, este trabalho propõe dois algoritmos de aprendizado. O primeiro é um algoritmo construtivo, o

qual tem por objetivo definir de forma automática o número de regras nebulosas necessárias para a modelagem da série temporal, de forma simultânea ao ajuste dos parâmetros durante o aprendizado *offline*. O modelo final obtido empregando este algoritmo de aprendizado é chamado de **C-FSM**. Este algoritmo e as condições para gerar e eliminar regras nebulosas durante o aprendizado são descritos no Capítulo 5.

O segundo algoritmo utiliza uma abordagem seqüencial, de tal forma que a estrutura do modelo é modificada a medida que os dados são apresentados, obtendo assim o modelo denominado de **A-FSM**. O Capítulo 6 descreve este algoritmo, assim como os operadores de geração e punição de regras nebulosas utilizados nesta abordagem.

Os resultados descritos no Capítulo 7 são compostos por conjunto de análises de séries temporais de natureza diversa. Primeiramente, os resultados obtidos para a série de Box-Jenkins são descritos, seguidos pelos resultados obtidos para uma série econômica correspondente a taxa de desemprego trimestral dos Estados Unidos (Tsay, 2005).

Este capítulo também apresenta alguns resultados de simulação obtidos para três séries financeiras, das onze séries que formaram o conjunto reduzido de dados utilizado na competição de previsão de séries temporais - *Artificial Neural Network & Computational Intelligence Forecasting Competition, 2007-NN3* - realizado como parte do *International Joint Conference on Neural Networks 2007* em Orlando, USA. Estes resultados incluem a análise de erros das previsões de um até dezoito passos à frente.

Além disso, os modelos propostos são aplicados na geração de modelos de previsão para as séries de vazões mensais e semanais do posto de Furnas, localizado na região sudeste do Brasil, ilustrando neste estudo todo o processo de modelagem da série temporal, considerando todas as etapas descritas assim como os diferentes procedimentos propostos para cada etapa.

Finalmente, o Capítulo 8 apresenta as conclusões do trabalho. Este capítulo também resume os principais objetivos atingidos, assim como propostas dos trabalhos base para pesquisas futuras.

2 *Pré-processamento e seleção de entradas*

2.1 Introdução

A especificação de um modelo de série temporal é um processo que começa na observação e análise dos dados e finaliza com a definição do modelo mais adequado para a série em estudo. Para tal, diversas etapas devem ser consideradas, como são o pré-processamento da série, a seleção de entradas, a natureza do modelo a utilizar e finalmente, o ajuste e otimização dos parâmetros do modelo, também chamada de etapa de estimação, o qual geralmente considera alguma métrica de desempenho de interesse na aplicação, para fins de validação.

Este capítulo apresenta as técnicas a utilizar nas duas primeiras etapas mencionadas: o pré-processamento da série e a seleção de entradas para os modelos de série temporal. As outras etapas são apresentadas no decorrer dos próximos capítulos.

O pré-processamento dos dados adotado neste trabalho é apresentado na Seção 2.2. O passo a seguir após a análise e o pré-processamento dos dados, é a seleção de entradas para o modelo de série temporal.

A seleção das entradas é fundamental para obter um modelo eficiente de previsão de séries temporais (Moon et al., 1995). No caso de modelos lineares, tais como os modelos de regressão ou modelos auto-regressivos, a seleção de entradas ou regressores define parcialmente a estrutura do modelo a utilizar, pois ainda é necessário continuar com a estimação dos parâmetros do modelo e finalmente, com a etapa de validação.

Já no caso dos modelos não-lineares, como redes neurais ou sistemas nebulosos, a determinação da estrutura do modelo precisa ainda da definição prévia de outros parâmetros tais como número de camadas e neurônios para redes neurais, ou número de regras, no caso dos sistemas nebulosos, além de outros parâmetros necessários na etapa de estimação. No entanto, as variáveis de entrada selecionadas para a construção dos modelos influencia todas

as próximas etapas, destacando a importância da escolha das variáveis de entradas.

De acordo com Guyon e Elisseeff (2003), os critérios e/ou métodos de seleção de entradas são classificados em três grupos: filtros, *wrappers* e *embedded*. Os critérios de seleção de entradas do tipo filtro são aqueles baseados apenas nos dados disponíveis e são independentes do modelo de série temporal a ajustar. Os *wrappers* são aqueles onde o modelo de série temporal é considerado como uma caixa preta e tem por objetivo a maximização de um índice de desempenho associado ao modelo ajustado utilizando um determinado subconjunto de variáveis candidatas a compor o vetor de entrada. Diversos modelos são ajustados utilizando diferentes subconjuntos de variáveis, os quais são construídos a partir de um conjunto contendo todas as possíveis variáveis de entrada. O subconjunto escolhido será aquele associado ao modelo com melhor desempenho. Por outro lado, os métodos de seleção de entradas do tipo *embedded* são aqueles onde a seleção de entradas é realizada de forma simultânea ao ajuste do modelo e é geralmente utilizado para certos tipos de máquinas de aprendizado. Diversos exemplos para cada uma das classes podem ser encontrados nos trabalhos descritos por Kohavi e John (1997) e por Guyon e Elisseeff (2003).

Uma forma de determinar as variáveis de entrada é utilizando os critérios de informação, tais como de informação *bayesiana* (Schwarz, 1978), de *Akaike* (Akaike, 1969), ou *Minimum Description Length*, os quais combinam a variância residual e a complexidade do modelo (seja a ordem do modelo linear ou o número de graus de liberdade no caso de um modelo não-linear) para estabelecer a necessidade de adequação do mesmo com o *princípio da parcimônia* (Haber e Unbehauen, 1990), (Ljung, 1999). Os métodos de seleção de entradas que utilizam estes critérios podem ser classificados como sendo do tipo *wrappers*.

A Seção 2.3 apresenta uma breve descrição do critério de Informação Bayesiana o qual é utilizado para comparação dos resultados obtidos na modelagem de séries de vazões, quando utilizado o critério de seleção de entradas proposto neste trabalho.

Este trabalho propõe uma metodologia de escolha de entradas para modelos de previsão, baseado em dois algoritmos: o algoritmo dos Falsos Vizinhos mais Próximos (FNN - *False Nearest Neighbors*) e o critério de Informação Mútua Parcial (PMI - *Partial Mutual Information*). Ambos os algoritmos são classificados como filtros, já que não é necessário definir a estrutura do modelo *a priori*, assim como nenhuma medida de desempenho associada a algum modelo de série temporal.

O primeiro, o algoritmo FNN, é apresentado na Seção 2.4.1. O critério de Informação Mútua Parcial (PMI - *Partial Mutual Information*) proposto inicialmente em (Sharma, 2000) é descrito na Seção 2.4.2, onde algumas modificações são apresentadas.

A abordagem proposta neste trabalho é descrita na Seção 2.4.3. A aplicação destes critérios e algoritmos será apresentada no Capítulo 7 deste trabalho, principalmente na análise de séries financeiras e séries de vazões afluentes mensais e semanais.

2.2 Pré-processamento dos dados

O pré-processamento dos dados, como foi mencionado no Capítulo 1, é uma etapa inicial e fundamental para a construção dos modelos de séries temporais. Existem diversas correntes neste sentido, já que, alguns trabalhos de pesquisa mostram-se a favor da transformação das séries temporais, enquanto outros defendem o esquema de trabalho baseados nos dados reais, ou seja, sem aplicar nenhuma transformação que retire, seja a tendência da série, a sazonalidade, ou alguma outra componente.

Como será visto neste capítulo, alguns critérios de seleção de variáveis - como é o caso do BIC e do PMI - trabalham com séries estacionárias. Por tal motivo, as séries que serão apresentadas no Capítulo 7 serão pré-processadas quando necessário, com a finalidade de obter a série estacionária. Para tal, é necessário analisar o histórico da série e verificar a existência de tendência e/ou sazonalidade. Existem diversos testes na literatura, assim como programas que auxiliam este tipo de análise.

Neste trabalho, a verificação da existência do componente tendência na série foi feita utilizando o teste de raízes unitárias. Uma descrição detalhada deste teste pode ser encontrada em (Hamilton, 1994) e em (Aguirre, 2004).

O componente de tendência pode ser retirado da série y^k aplicando o operador de diferença, tantas vezes quanto for necessário. O número de vezes que é necessário utilizar este operador é determinado por meio do teste de raiz unitária. Este teste, conhecido também como teste aumentado de Dickey-Fuller (ADF - *Augmented Dickey-Fuller teste*) (Hamilton, 1994) é aplicado com a ajuda do software *Econometric Views* (Quantitative Micro Software, 1995). O teste ADF fornece as estatísticas conhecidas como estatísticas t , as quais determinam a existência de raízes unitárias em um modelo de regressão de primeira diferença da série $\Delta y^k = y^k - y^{k-1}$ versus a série defasada no tempo y^{k-1} , diferenças defasadas no tempo e/ou um termo constante e uma tendência em k (Quantitative Micro Software, 1995).

Estas estatísticas são em outras palavras, os coeficientes associados a cada componente incluído no modelo de regressão ajustado, principalmente, ao coeficiente de y^{k-1} . Uma vez que o modelo é ajustado e definido, procede-se o cálculo das raízes do polinômio obtido. A existência de raízes unitárias indicará que a primeira transformação realizada, neste caso, a

primeira diferença já que o modelo foi construído em Δy^k , não foi suficiente para eliminar por completo o componente de tendência. Por outro lado, se o polinômio não contém raízes unitárias, significa que a primeira diferença foi suficiente para retirar o componente de tendência da série. O processo é repetido até que não seja mais necessário diferenciar a série.

Assim, se o coeficiente de y^{k-1} é significativamente diferente de zero, então, a hipótese de que y contém uma raiz unitária é rejeitada e a hipótese de que y é estacionária é aceita. Da mesma forma, valores negativos das estatísticas t rejeitam a hipótese de existir componentes de tendência, sugerindo que a série é estacionária.

Os limiares para estas estatísticas são definidos baseados na quantidade de dados disponíveis, assim como no modelo ajustado. Se estas estatísticas são inferiores em valor absoluto aos limiares calculados, então, a hipótese de existência de raízes unitárias não pode ser rejeitada, sugerindo uma série não estacionária.

No caso da série temporal ser uma série com sazonalidade mensal (período de 12 meses), a transformação para a sua versão estacionária pode ser efetuada da seguinte forma:

$$z^k(m) = \frac{y^k(m) - \mu(m)}{\sigma(m)} \quad (2.1)$$

onde $z^k(m)$ é a série mensal estacionária no mês m , para uma série temporal sem tendência; $y^k(m)$ é a observação no instante k e mês m , $\mu(m)$ é a média histórica mensal e $\sigma(m)$ é o desvio padrão mensal da série.

Desta forma, antes de selecionar as entradas para um modelo de série temporal, elimina-se o componente sazonal da série, para logo em seguida, verificar por meio do teste de raízes unitárias, o número de vezes que a série deve ser diferenciada obtendo, finalmente, a transformação da série original para uma série estacionária.

2.3 Critério de informação de Bayes - BIC

Nesta seção é descrito o critério de informação de Bayes - BIC (Schwarz, 1978). Este critério de seleção de entrada é baseado em métricas de correlação linear e está diretamente vinculada às estimativas das ordens ótimas de modelos de regressão.

Seja z_t , $t = 0, 1, \dots$ uma série temporal estacionária. O critério de informação de Bayes, ou critério BIC é definido como:

$$BIC(p) = N \cdot \ln(\hat{\sigma}_p^2) + p \ln(N); \quad (2.2)$$

sendo N o número de observações, p a ordem ou número de atrasos do modelo no caso de um modelo auto-regressivo e o grau de liberdade para outros casos. $\hat{\sigma}_p^2$ a estimativa da variância do ruído branco a_t para um modelo ajustado de ordem p .

A partir de uma ordem inicial, procede-se o ajuste do modelo. O processo é repetido para ordens superiores, sendo o valor do critério BIC calculado para cada ordem p . O menor valor do critério BIC corresponde à ordem do modelo escolhida. Nota-se que este critério leva em conta tanto a variância do erro como o número de parâmetros a ser ajustado para a escolha do modelo. Além deste critério, sugere-se a análise dos resíduos para que o modelo ajustado seja o mais adequado para a previsão da série temporal.

Diferentemente de outros critérios da literatura, que tendem a superestimar a ordem do modelo para um número grande de amostras, o critério BIC favorece a escolha de modelos com o menor número de parâmetros possível, penalizando caso ocorra o contrário.

Como pode ser observado, este critério depende da estrutura do modelo a ser escolhida, ou seja, pertencem ao tipo de critério paramétrico de seleção de entradas, pois a estrutura geral do modelo deve ser definida e ajustada para que a ordem do modelo seja escolhida.

Por outro lado, o critério BIC pode ser utilizado como medida de desempenho de modelos de séries temporais de natureza diversa, linear ou não-linear, desde que a estrutura e parâmetros do modelo sejam definidos previamente ao cálculo do BIC ou tenha-se como objetivo a seleção de modelos com parcimônia, por exemplo, na determinação do número de neurônios para a camada oculta de uma rede neural.

Existe uma segunda abordagem na literatura, que engloba outros critérios e técnicas de seleção de entradas, que não precisam definir *a priori* o tipo de modelo a ser utilizado, ou seja, são baseados apenas em dados e não em um modelo já definido. Estas técnicas são conhecidas como técnicas não-paramétricas de seleção de entradas.

2.4 Técnicas não paramétricas de seleção de entradas

Os métodos não paramétricos de seleção de entradas são aqueles que dependem apenas dos dados disponíveis, devido à utilização de medidas estatísticas de dependência entre as variáveis consideradas. Ou seja, não é necessário definir uma estrutura de modelo de série

temporal e ajustar os seus parâmetros, para então, escolher o número de variáveis de entrada que são relevantes para o modelo de série temporal. Os filtros são uma classe de métodos de seleção de entradas que também se encaixam como técnicas não paramétricas.

As possíveis entradas são construídas com base nos dados disponíveis, e uma medida estatística é avaliada para todo o conjunto. As variáveis com maiores índices de relevância são mantidas, e as demais são descartadas já que, pelo índice estatístico escolhido, estas tendem a apresentar uma menor capacidade de predição do modelo de série temporal a ser construído.

Uma métrica estatística conhecida na teoria de informação é a informação mútua (MI - *mutual information*) (Cover e Thomas, 1991), que mede a quantidade de informação a partir de uma variável \mathbf{x}_1 , na presença de \mathbf{x}_2 , e vice-versa, podendo ser utilizada para medir a dependência entre variáveis aleatórias (Reyhani et al., 2005). No entanto, este critério não é capaz de eliminar variáveis redundantes.

Geralmente, como é o caso dos trabalhos descritos em (Bonnlander e Weigend, 1994) e em (Reyhani et al., 2005), a partir de um conjunto de entradas candidatas, é necessário avaliar todas as combinações possíveis de entradas para escolher o melhor conjunto que maximize o critério MI. No entanto, embora os modelos de previsão forneçam um desempenho satisfatório, o processo de seleção de entradas torna-se complexo em termos computacionais, existindo sempre a possibilidade de variáveis redundantes no conjunto escolhido.

Uma alternativa computacionalmente mais simples é o algoritmo dos Falsos Vizinhos mais Próximos (FNN) descrito a seguir.

2.4.1 Algoritmo dos falsos vizinhos mais próximos - FNN

O algoritmo de falsos vizinhos mais próximos (FNN), foi desenvolvido para determinar o menor número de atrasos possíveis p a serem utilizados, com a finalidade de determinar a saída y^t diretamente de um conjunto de dados que representa a dinâmica do sistema.

A reconstrução do espaço de estado é baseada no Teorema de Imersão de Takens (Takens, 1981). O Teorema de Takens (Takens, 1981) prova que, todo sistema dinâmico determinístico pode ser representado por um número mínimo de atrasos $p > 2d$, ou seja:

$$y^t = G[y^{t-\tau}, y^{t-2\tau}, \dots, y^{t-p\tau}] \quad (2.3)$$

sendo d a dimensão real do espaço de estados, G a função que descreve o sistema e τ um

número inteiro positivo. Ou seja, o teorema permite reconstruir um espaço de estado p -dimensional, similar ao espaço de estado original d -dimensional, a partir da variável medida.

O algoritmo FNN - que foi inicialmente proposto com o objetivo de determinar a dimensão de imersão mínima para reconstruir o espaço de estados, de tipos de sistemas representados pela Equação (2.3), a partir de dados observados (Abarbanel e Kennel, 1993) - é baseado no seguinte fato (Rhodes e Morari, 1997). Dada uma dimensão de imersão p muito pequena, tem-se que, pontos próximos no espaço p -dimensional podem parecer próximos apenas por causa da projeção e não devido à dinâmica do sistema. Assim, embora estes pontos estejam próximos, as suas saídas podem ser bastante diferentes. Estes pontos que aparentam ser próximos são denominados de “falsos vizinhos”. Se p é suficientemente grande para representar a dinâmica do sistema, então, os pontos próximos sempre terão saídas próximas. Assim, avalia-se se um vizinho é “verdadeiro” ou “falso” apenas em virtude da projeção do sistema em uma determinada dimensão.

Como um exemplo de falso vizinho devido à falta de uma variável, ou seja, a escolha de uma dimensão p inferior ao necessário, considera-se:

$$y_f = x_1 \quad \text{e} \quad y = x_1 + x_2$$

sendo y_f representado pelos pontos em azul e y representado pelos pontos em vermelho (ver Figura 1). As variáveis x_1 e x_2 são definidas tal que:

$$x_1 = [0, 10 \quad 0, 15 \quad 0, 75] \quad \text{e} \quad x_2 = [0, 10 \quad 1, 0 \quad 0, 10]$$

Como pode ser observado na Figura 1, se apenas a variável x_1 for considerada, o ponto 2 estará mais próximo ao ponto 1, sendo então o ponto 2 considerado como vizinho do ponto 1 (y_f), e vice-versa. Além disso, observa-se que o ponto 3 encontra-se distante dos outros dois pontos.

No entanto, se x_1 e x_2 são consideradas como variáveis, observa-se que os pontos 1 e 2 encontram-se distantes um do outro. Além disso, para $p = 2$ o ponto 3 continua distante dos outros dois pontos. Desta forma, para $p = 1 \times (y_f)$, tendo como única entrada x_1 , o ponto 2 será um falso vizinho do ponto 1. Já se $p = 2 \times (y)$ esta situação é corrigida. Neste caso, $p = 2$ mostra-se como um número de atrasos mais adequado para modelar y .

Dada uma dimensão de imersão p , para cada ponto $\mathbf{z}_p^k = [y^{k-\tau}, \dots, y^{k-p\tau}]$, com $k = 1, \dots, N$, o ponto mais próximo \mathbf{z}_p^j será denominado de vizinho. O critério para determinar se o vizinho

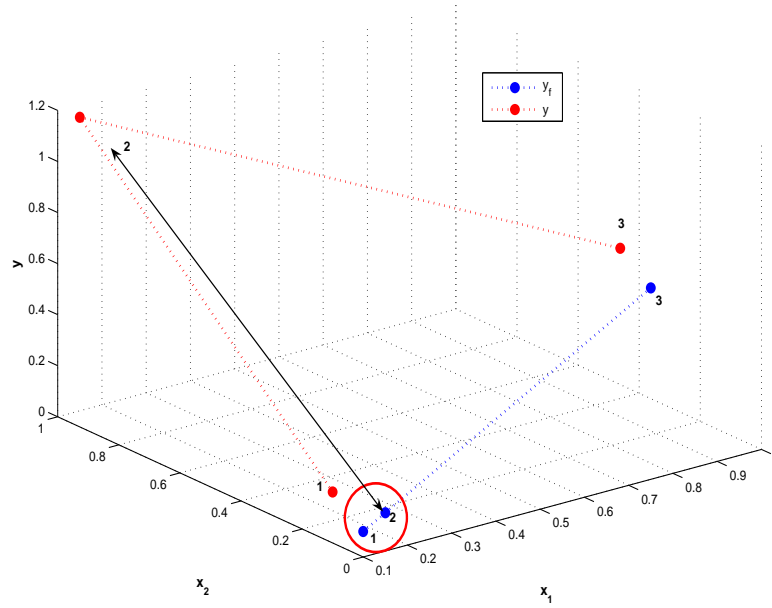


Figura 1: Exemplo de um falso vizinho.

é verdadeiro ou falso é definido por:

$$\frac{|y^k - y^j|}{\|\mathbf{z}_p^k - \mathbf{z}_p^j\|_2} \leq R \quad (2.4)$$

sendo $\|\cdot\|_2$ a distância euclidiana e R um limiar previamente definido, com valores entre 10 e 30. (Rhodes e Morari, 1997) mostram que, para valores de R neste intervalo, o número de falsos vizinhos se mantém quase constante. Nesse mesmo trabalho, critérios teóricos de seleção deste limiar podem ser encontrados. Se a Equação (2.4) é satisfeita, então, \mathbf{z}_p^j é um verdadeiro vizinho. Caso contrário, \mathbf{z}_p^j é um falso vizinho.

A cada iteração o número de falsos vizinhos é calculado e o processo continua, sendo p incrementado a cada iteração, até que o número de falsos vizinhos seja pequeno ou próximo a zero.

2.4.1.1 Algoritmo FNN

O algoritmo FNN pode ser resumido nos seguintes passos:

Algoritmo 1 FNN

Definir uma dimensão de imersão inicial p ;

Passo 1. Para cada ponto $\mathbf{z}^k = [y^{k-\tau}, \dots, y^{k-p\tau}]$, com $k = 1, \dots, N$, determinar o vizinho \mathbf{z}^j .

Passo 2. Verificar o critério definido pela Eq. (2.4).

Passo 3. Após determinar se os vizinhos são verdadeiros ou falsos para cada ponto, calcular o percentual de pontos com falsos vizinhos.

Passo 4. Se o número de falsos vizinhos é próximo a zero ou um valor pequeno aceitável, encerrar o processo, e definir os primeiros p atrasos como sendo as entradas do modelo. Caso contrário, fazer $p = p + 1$ e voltar ao passo 1.

Embora este algoritmo seja simples, o algoritmo FNN ainda não é capaz de escolher entradas sem o risco de redundância, já que a escolha da “janela de previsão” é feita de forma incremental e seqüencial, ou seja, o número de variáveis, é incrementado consecutivamente até o número de falsos vizinhos atingir um valor próximo a zero, obtendo assim, modelos com complexidade maior que a necessária e comprometendo o desempenho do modelo de previsão.

Para eliminar as variáveis menos relevantes do conjunto inicial gerado pelo algoritmo FNN, neste trabalho é proposta a aplicação do Critério de Informação Mútua Parcial a partir das informações obtidas pelo algoritmo FNN.

2.4.2 Critério de informação mútua parcial - PMI

O critério de informação mútua, como já foi mencionado, surge da teoria de informação, embora seja aplicado em outras áreas tais como reconhecimento de padrões, processamento de imagens, seleção e identificação de variáveis de entrada para modelos não-paramétricos (Bonnlander e Weigend, 1994), (Zaffalon e Hutter, 2002), (Martins, 2004).

A informação mútua é uma medida bastante usada para análise de dependência estocástica de variáveis aleatórias discretas (Cover e Thomas, 1991), (Soofi, 2000). Este critério fornece uma medida do grau de dependência entre variáveis, sendo um indicador importante na análise de séries temporais. Assim, se duas variáveis são independentes, o critério de informação mútua será zero; se duas variáveis são fortemente dependentes, a informação mútua terá um valor alto.

A informação mútua pode também ser considerada como uma medida da quantidade de informação armazenada em uma variável com relação a outra, sendo por este motivo, interessante na escolha de entradas para modelos de sistemas não-lineares, principalmente de

redes neurais, onde o processo de aprendizado pode ser considerado como um mecanismo de extração de conhecimento (Bonnlander e Weigend, 1994), (Zheng e Billings, 1995).

O critério de informação mútua MI (*Mutual Information*) entre duas variáveis contínuas x e y amostradas em tempo discreto é definido como:

$$MI = \frac{1}{N} \sum_{i=1}^N \log_e \left[\frac{f_{xy}(x_i, y_i)}{f_x(x_i)f_y(y_i)} \right] \quad (2.5)$$

sendo (x_i, y_i) o i -ésimo par de dados do conjunto de amostras, com $i = 1, \dots, N$, $f_{xy}(x, y)$ a função de probabilidade conjunta (*Probability Density Function* - PDF) de x e y ; e $f_x(x)$ e $f_y(y)$ são as funções de probabilidade marginais de x e y , respectivamente. Se x e y são independentes, a função de probabilidade conjunta será igual ao produto das probabilidades marginais, sendo o valor do critério MI igual a 0. Caso x e y sejam dependentes, o valor de MI será maior que 0.

Existem diversas maneiras de aproximar as funções de probabilidade marginais e conjuntas. Em (Fraser e Swinney, 1986), são empregados histogramas de frequência como médias da estimativa das probabilidades.

Uma variante desta técnica é utilizada em (Bonnlander e Weigend, 1994), onde a técnica de aproximação das funções de densidade de probabilidade é chamada de *Equal mass binning*. Diferente da técnica de aproximação tradicional por histogramas, cada partição do espaço unidimensional é realizada com base na quantidade de amostras, que deve ser igual para cada partição. A probabilidade associada a cada partição, ou *bin*, será igual ao número relativo de amostras dividido pelo tamanho da partição respectiva.

Neste trabalho, foram utilizadas aproximações baseadas em funções de *kernel*, tanto para as funções de probabilidades marginais como para as funções de probabilidade conjunta, por ser uma maneira mais eficiente e robusta como mostrado em (Sharma, 2000). A aproximação das funções de probabilidade é apresentada a seguir.

2.4.2.1 Aproximação das funções de probabilidades marginal e conjunta

Seja N o número de pares de dados entrada-saída $[\mathbf{x}^k, y^k]$, com $k = 1, \dots, N$. A função de probabilidade marginal de uma variável \mathbf{x} unidimensional pode ser aproximada por meio de estimadores não paramétricos de *kernel*, dada por:

$$\hat{f}(\mathbf{x}) = \frac{1}{N\lambda} \sum_{i=1}^N K\left(\frac{x - x_i}{\lambda}\right) = \frac{1}{N} \sum_{i=1}^N K_\lambda(x - x_i) \quad (2.6)$$

sendo que $K_\lambda(t)$ é denominada de função de *kernel*, e λ é a largura de banda ou parâmetro de dispersão.

Existem diversas funções de *kernel* (Silverman, 1986). Dentre estas, (Bowden et al., 2005) utiliza a função de distância absoluta ou *city-block*. A vantagem de utilizar este tipo de função está no fato de ser simples em termos computacionais, assim como não precisar assumir nenhum tipo de distribuição dos dados como é feito em (Sharma, 2000) ou em (Akaho, 2002).

A função de probabilidade marginal de \mathbf{x} , utilizando a distância absoluta, é definida por:

$$\hat{f}_x(\mathbf{x}) = \frac{1}{N(2\lambda)^p} \sum_{i=1}^N \prod_{j=1}^p \exp^{-|\mathbf{x}_j - \mathbf{x}_{ij}|/\lambda} \quad (2.7)$$

isto é,

$$\hat{f}_x(\mathbf{x}) = \frac{1}{N(2\lambda)^p} \sum_{i=1}^N \exp \left[-\frac{1}{\lambda} \sum_{j=1}^p |\mathbf{x}_j - \mathbf{x}_{ij}| \right] \quad (2.8)$$

As Equações (2.7) e (2.8) surgem a partir da Equação (2.6), como uma adaptação para o caso de \mathbf{x} ser multidimensional e utilizando como função de *kernel* a função *city-block*. Na Equação (2.8), p é a dimensão do vetor \mathbf{x} e $\hat{f}_x(\mathbf{x})$ a aproximação da função de probabilidade marginal de \mathbf{x} . O parâmetro λ é calculado como:

$$\lambda = \left(\frac{4}{p+2} \right)^{1/(p+4)} N^{-1/(p+4)} \quad (2.9)$$

Embora a definição de λ dada na Equação (2.9) tenha sido derivada para dados com distribuição normal, esta equação é amplamente utilizada na literatura por ser uma definição simples e eficiente como mostrado em (Bowden et al., 2005).

A Figura 2 ilustra um exemplo de aproximação de função densidade de probabilidade bi-variável, de x_1 e x_2 , seguindo o procedimento adotado em (Bowden et al., 2005). A distribuição de cada uma das variáveis é ilustrada na Figura 2-(a) e 2-(b), por meio dos respectivos histogramas unidimensionais, com um total de 2000 pares de dados gerados aleatoriamente considerando uma distribuição normal no intervalo $[-0,5, 0,5]$.

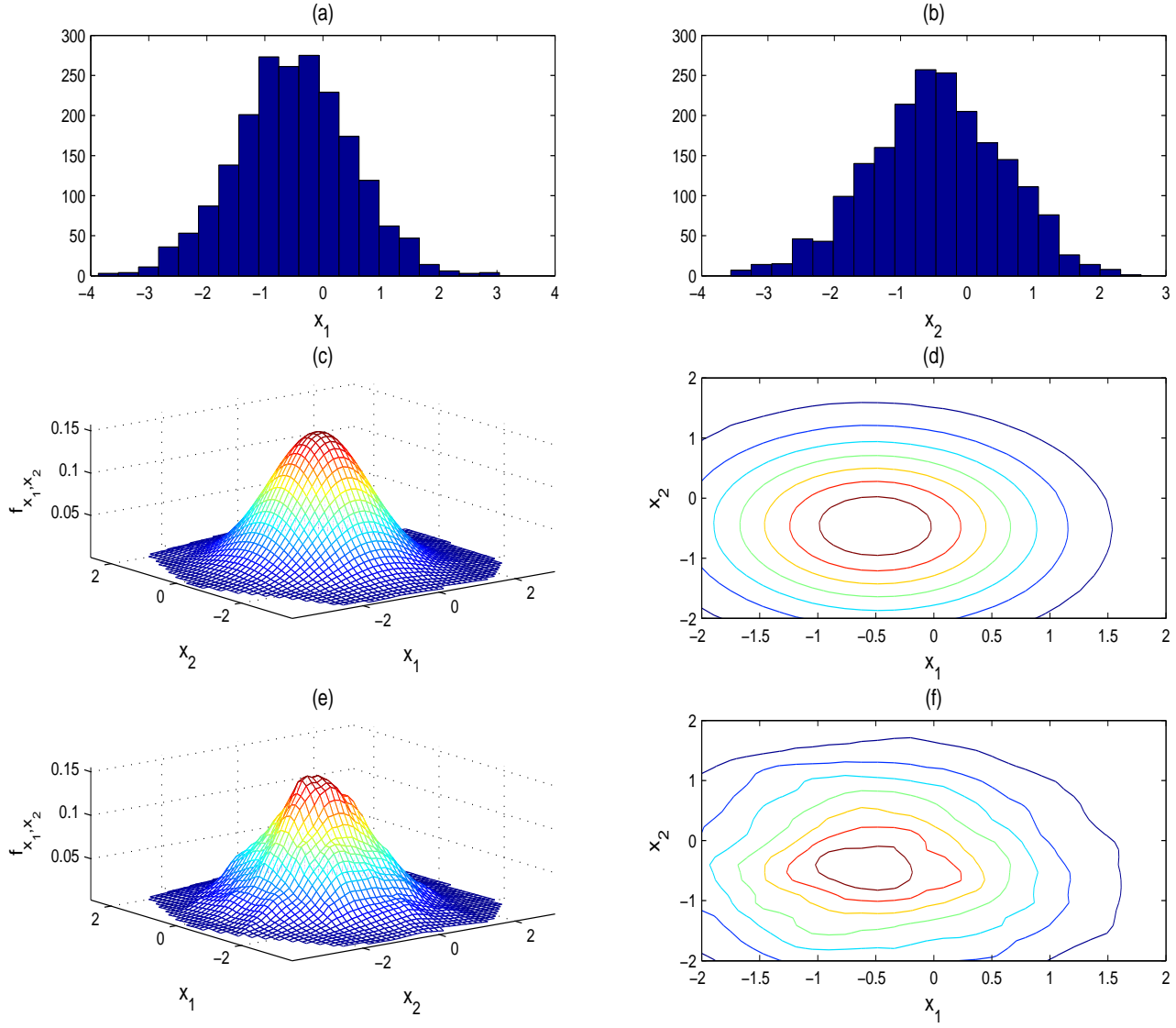


Figura 2: Aproximação de distribuições de probabilidade utilizando diferentes funções de *kernel*.

A função de distribuição de probabilidade gerada é ilustrada na Figura 2-(c), a qual foi construída utilizando a função de distribuição gaussiana bi-variável dada por:

$$f_{x_1, x_2} = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}\sqrt{1-\rho^2}} \exp -0.5X(1-\rho^2) \quad (2.10)$$

na qual X é definido da seguinte maneira:

$$\mathcal{X} = \frac{(x_1 - \mu_{x_1})^2}{\sigma_{x_1}^2} + \frac{(x_2 - \mu_{x_2})^2}{\sigma_{x_2}^2} - 2\rho \frac{(x_1 - \mu_{x_1})(x_2 - \mu_{x_2})}{\sigma_{x_1}\sigma_{x_2}}$$

com ρ sendo o coeficiente de correlação entre x_1 e x_2 , σ_{x_1} e σ_{x_2} são os desvios padrão de x_1 e x_2 respectivamente, e μ_{x_1} e μ_{x_2} são os valores médios das variáveis.

Assim, a função de densidade de probabilidade f_{x_1, x_2} calculada pela Equação (2.10) é ilustrada na Figura 2-(c). A Figura 2-(d) ilustra o diagrama de curvas de nível obtido para esta estimativa.

A Figura 2-(e) ilustra a aproximação da função de densidade de probabilidade \hat{f}_{x_1, x_2} construída a partir das Equações (2.8) e (2.9), utilizando a função de kernel do tipo *city-block*. O diagrama de curvas de nível associado a esta aproximação é apresentado na Figura 2-(f). Como pode ser observado, a estimativa utilizada neste trabalho, fornece uma boa aproximação, embora as curvas de nível não sejam perfeitamente concêntricas como apresentado na Figura 2-(d). O coeficiente de correlação entre a função de probabilidade bi-variável f_{x_1, x_2} e a sua estimativa \hat{f}_{x_1, x_2} foi igual a 0,99.

Da mesma forma, a função de probabilidade conjunta de \mathbf{x} dado y , com y sendo a saída unidimensional, é definida por (Akaho, 2002):

$$\hat{f}_{xy}(\mathbf{x}, y) = \frac{1}{N(\lambda)^{p+1}} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{\lambda}\right) K\left(\frac{y - y_i}{\lambda}\right) \quad (2.11)$$

ou seja:

$$\hat{f}_{xy}(\mathbf{x}, y) = \frac{1}{N(2\lambda)^{p+1}} \sum_{i=1}^N \exp\left[-\frac{1}{\lambda} s_i\right] \quad (2.12)$$

sendo s_i igual a:

$$s_i = \sum_{j=1}^p |\mathbf{x}_j - \mathbf{x}_{ij}| + |y - y_i| \quad (2.13)$$

Para ilustrar a aproximação das funções de probabilidades usando a função de *kernel* no cálculo do MI, considera-se o seguinte modelo:

$$y = x^4 + e_1 \quad x = \sin(2\pi t/T) + e_2 \quad (2.14)$$

sendo $T = 20$, $t = 1, \dots, 200$ e e_1 e e_2 ruídos com distribuição normal de média zero e desvio padrão igual a 0,1. A Figura 3 e a Equação (2.14) ilustram de forma clara uma relação não-linear entre a variável independente x e a variável dependente y .

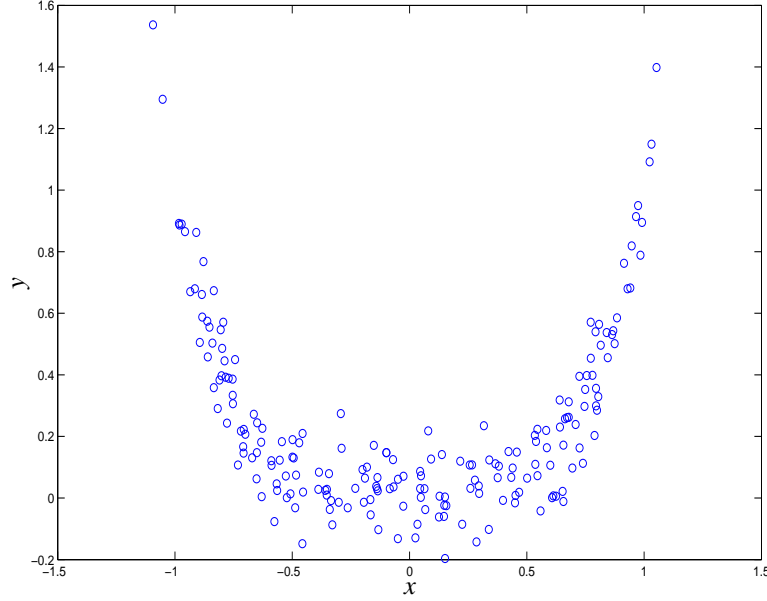


Figura 3: Exemplo de dependência não-linear entre duas variáveis.

O coeficiente de correlação linear entre estas duas variáveis é de 0,0032, indicando que praticamente não existe relação linear entre ambas as variáveis, isto devido à relação entre x e y ser não-linear. Aplicando o critério de informação mútua (Equação (2.5)) nas amostras do modelo da Equação (2.14), obtém-se um valor de MI igual a 0,4199. A Tabela 1 mostra os valores obtidos para os critérios de correlação e para o critério MI, com o correspondente limiar. Este limiar foi calculado tendo como base o percentil de 95%, indicando que a dependência entre as duas variáveis é forte, se o valor obtido for maior que este limiar, com uma probabilidade de erro de 5%.

Observa-se na Tabela 1 que, enquanto o coeficiente de correlação indica independência entre as variáveis, o critério de informação mútua indica dependência entre x e y , obtendo um valor de 0,4199 superior ao limiar igual a 0,1154.

Em (Sharma, 2000), é proposto o critério de informação mútua parcial (PMI) o qual é uma medida da redução da incerteza em y devido ao conhecimento de x , ou seja, o critério PMI mede a informação mútua entre a variável independente x e a variável dependente y , condicional a um conjunto de entradas \mathbf{z} selecionado inicialmente. Dado que \mathbf{z} existe, é

Tabela 1: Comparação do critério de informação mútua e do coeficiente de correlação.

Critério	Valor	Percentil (95%)
MI	0,4199	0,1154
Correlação	0,0032	0,0965

necessário extrair a influência desse conjunto em relação aos atrasos ainda não selecionados, para calcular a real contribuição de x . Assim, a Equação (2.5) pode ser reformulada da seguinte maneira:

$$\text{PMI} = \frac{1}{N} \sum_{i=1}^N \log_e \left[\frac{f(x'_i, y'_i)}{f_{x'}(x'_i) f_{y'}(y'_i)} \right] \quad (2.15)$$

na qual,

$$x'_i = x_i - E(x_i|\mathbf{z}) \quad \text{e} \quad y'_i = y_i - E(y_i|\mathbf{z}) \quad (2.16)$$

sendo x'_i e y'_i os valores residuais correspondentes ao i -ésimo par de dados e $E(\cdot)$ o valor esperado condicional associado. As definições dadas na Equação (2.16) garantem que as variáveis x' e y' representem a informação restante, uma vez que o efeito das entradas escolhidas presentes no conjunto observado é considerado.

Como pode ser observado na Equação (2.16), para o cálculo do PMI, é necessário estimar o valor esperado $E(\cdot)$, para assim, estimar x' e y' .

Em (Sharma, 2000) foi assumido que os dados seguem uma distribuição normal, independente do sistema. Já no trabalho de Bowden (2003), foi proposta a utilização de uma rede neural de regressão (Specht, 1991), para aproximar os valores condicionais esperados, assim como a utilização da distância *city-block* ou diferença absoluta para a aproximação das funções de probabilidades marginais e conjuntas, tornando o tempo de execução do algoritmo menor quando comparado com a metodologia proposta em (Sharma, 2000).

Este trabalho propõe a utilização da função de distância *city-block* ou diferença absoluta, tanto para a aproximação das funções de probabilidades marginais e conjuntas, como na aproximação dos valores esperados condicionais, por meio de regressores de *kernel* (Scott,

1992), necessários no cálculo do critério PMI (Luna et al., 2006b). A vantagem de utilizar a distância *city-block*, em combinação com as funções de *kernel*, é que não é necessário conhecer a distribuição de probabilidade dos dados. Além disto, neste trabalho é utilizada a distância absoluta na aproximação dos valores esperados condicionais, diminuindo assim a complexidade das aproximações feitas em (Sharma, 2000), assim como a necessidade de ajustar um modelo de rede neural de regressão a cada iteração, como feito em (Bowden et al., 2005). Assim, o valor esperado é aproximado por um modelo de regressão não paramétrica de *kernel*, conhecido como Estimador de Nadaraya-Watson (Scott, 1992).

2.4.2.2 Aproximação do valor esperado condicional

Seja $\hat{r}(x)$ a aproximação do valor esperado condicional $E(Y|X = x)$, definida da seguinte forma:

$$\hat{r}(x) = \sum_{i=1}^N w_{\lambda_x}(x, x_i) y_i \quad (2.17)$$

sendo,

$$w_{\lambda_x}(x, x_i) = \frac{K_{\lambda_x}(x - x_i)}{\sum_{j=1}^N K_{\lambda_x}(x - x_j)} \quad (2.18)$$

na qual, $K_{\lambda_x}(x - x_i)$ é a função de *kernel* para x . Nesta aproximação também foi empregada a função *city-block*. Este critério foi adotado por ser mais simples de implementar, quando comparado com a aproximação feita em (Sharma, 2000) e em (Bowden, 2003).

2.4.2.3 Determinação do limiar de confiança para o PMI

Existem duas maneiras de finalizar o processo de seleção de entradas. A primeira consiste na construção de um teste de hipótese utilizando p seqüências diferentes de x em relação a y , por meio de um reordenamento ou *bootstrapping* da variável independente, e obtendo o respectivo vetor de PMIs. Ou seja, o limiar de confiança para determinar se uma entrada é relevante para modelar o sistema é calculado considerando como hipótese nula a independência entre a variável de entrada e a variável de saída.

Os PMIs obtidos são reordenados de forma ascendente e o valor correspondente ao γ -th percentil será o limiar para rejeitar ou não rejeitar a hipótese de independência entre as

variáveis. Dessa forma, a hipótese de independência entre as variáveis será rejeitada caso a entrada avaliada possua um PMI superior ao nível de significância ($\alpha = 1 - \gamma$) estabelecido, ou seja, a entrada será considerada relevante com uma possibilidade de $\alpha\%$ destas serem realmente independentes. Resultados do algoritmo proposto na modelagem de séries temporais utilizando redes neurais recorrentes encontram-se em (Luna et al., 2006a), onde adotou-se $p = 100$ seqüências, um percentil $\gamma = 95\%$ e, portanto, um nível de significância $\alpha = 5\%$.

O segundo critério de parada consiste em estabelecer um mínimo de PMI para não rejeitar a entrada avaliada. Caso o valor do PMI seja menor a este valor mínimo estabelecido, a entrada é rejeitada. Neste caso, uma visualização gráfica facilita a análise, como será visto no Capítulo 7. Este critério de parada é factível devido à natureza monotonicamente decrescente do PMI a medida que as entradas são selecionadas.

Em termos gerais, tanto o limiar de confiança como o valor mínimo pré-definido para o PMI refletem o rigor com o qual as entradas serão selecionadas. Maior o nível de significância na escolha de entradas, mais rigorosa será a seleção das mesmas, já que estas terão que possuir um valor mais alto de PMI para serem escolhidas. Ou seja, se o valor de α é baixo, tem-se o risco de escolher entradas pouco relevantes para a modelagem no final do processo. Por outro lado, se α é muito alto, entradas importantes podem estar sendo rejeitadas, prejudicando a modelagem da série.

2.4.2.4 Algoritmo PMI

O algoritmo de seleção de entradas baseado no critério PMI pode ser resumido como segue:

Algoritmo 2 PMI

Construir um conjunto inicial de possíveis entradas denotado por \mathbf{z}^* . Definir um conjunto de entradas \mathbf{z} inicialmente vazio.

Passo 1. Calcular o critério PMI (Equação (2.15)) para cada possível entrada em \mathbf{z}^* e condicionado à existência de variáveis já selecionadas contidas em \mathbf{z} ;

Passo 2. Identificar a variável com maior valor de PMI, denotada por \mathbf{x}^* ;

Passo 3. Construir o teste de hipótese e/ou definir o limiar para o PMI de \mathbf{x}^* . Se o PMI de \mathbf{x}^* é maior que o limiar definido, então excluir \mathbf{x}^* de \mathbf{z}^* e acrescenta-la ao conjunto \mathbf{z} . Senão, ir ao passo 5;

Passo 4. Voltar ao passo 1;

Passo 5 Fim do processo.

2.4.3 Abordagem combinada: algoritmo FNN-PMI

Neste trabalho é proposta uma abordagem que combina os algoritmos FNN-PMI com o objetivo de aproveitar as vantagens dos dois algoritmos de seleção de entradas descritos nas seções anteriores.

A principal vantagem que a abordagem combinada apresenta é a definição automática do conjunto inicial a ser avaliado pelo algoritmo PMI (devido ao algoritmo FNN), assim como a capacidade de captar relações não-lineares existentes entre as variáveis de entrada e saída (devido ao algoritmo PMI). Assim, a combinação de ambos os algoritmos mencionados fornece em geral, um conjunto de variáveis mais compacto, quando comparado ao conjunto fornecido pelos algoritmo FNN de maneira independente, e através de um processo menos demorado, quando comparado ao processo manual de seleção de entradas utilizando apenas o algoritmo PMI.

Assim, o conjunto inicial de entradas candidatas \mathbf{z}^* a ser utilizado pelo algoritmo PMI será definido pelo algoritmo FNN e o conjunto final de variáveis a utilizar para construir o vetor de entrada para os modelos de série temporal é determinado pelo algoritmo PMI.

Em resumo, a abordagem combinada pode ser descrita nos seguintes passos:

Algoritmo 3 FNN-PMI

Passo 1. Aplicar o algoritmo FNN sobre a série de dados para obter um conjunto inicial de entradas \mathbf{z}^* ;

Passo 2. Aplicar o algoritmo PMI, sob o conjunto inicial de possíveis entradas, fornecido no Passo 1.

2.5 Resumo

Este capítulo mostrou uma breve descrição das técnicas utilizadas no pré-processamento dos dados, para o processo de seleção de variáveis e de modelagem em si.

Além disso, este capítulo apresentou duas técnicas não paramétricas da literatura utilizadas na seleção de variáveis de entrada para modelos de séries temporais. A vantagem observada nestes algoritmos é a capacidade de lidar com dependências não-lineares entre as variáveis. Além disso, este capítulo descreve uma abordagem combinada que tem por objetivo aproveitar as vantagens de ambos os algoritmos descritos.

3 *Modelos não lineares de séries temporais baseados em regras nebulosas*

3.1 Revisão bibliográfica

Nos últimos anos, a combinação de redes neurais com sistemas nebulosos em modelos integrados, que herdaram características destas duas abordagens tem tido sucesso em várias aplicações como controle e processamento de sinais (Jang, 1993), medicina (Mitra e Hayashi, 2000), séries temporais (Ballini, 2000), (Luna et al., 2005) entre outras.

Um sistema nebuloso é essencialmente um sistema baseado em regras lingüísticas do tipo se-então. Estas regras são as responsáveis por representar o conhecimento adquirido pelo especialista, ou extraído de dados na forma de variáveis de entrada e saída ao sistema. Dado que não existe uma maneira formal e efetiva de extração do conhecimento, a análise dos dados disponíveis para a extração de informação pelo especialista, com o objetivo de determinar o número de regras adequadas e assim, definir a estrutura do sistema nebuloso, torna-se uma tarefa difícil de ser resolvida (Er e Wu, 2002).

Geralmente, sistemas baseados em regras nebulosas do tipo se-então são compostos por um mecanismo de inferência nebulosa, uma base de regras e um *defuzzificador*, o qual é o responsável pelo processamento das saídas locais de cada uma das regras nebulosas (Luna et al., 2003). Já o mecanismo de inferência é o encarregado de mapear a transformação nebulosa do espaço de entrada para o espaço de saída, utilizando por sua vez, a base de regras que compõe o sistema.

Existem diversas abordagens para a construção de sistemas baseados em regras nebulosas, sendo que, cada uma pode obter uma estrutura diferente para o mesmo sistema, com resultados e desempenho similares, o que dificulta a construção de modelos e definição da base de regras.

Uma técnica bastante divulgada para construir uma base de regras nebulosa é a partição nebulosa do espaço de entrada em M sub-regiões, sendo que, cada uma das regras nebulosas é alocada para cada uma destas sub-regiões. Desta forma, o tamanho da base de regras, que define indiretamente a estrutura do sistema, não é definido apenas pelo número de entradas ao sistema, mas também pelo número de partições do espaço de entrada.

Em boa parte das propostas de sistemas nebulosos, como os descritos em (Lee et al., 1995), (Caminhas et al., 1999) ou em (Lee e Teng, 2000), a dependência da estrutura do modelo com o número de partições e o número de entradas gera um problema já conhecido na comunidade científica com o nome de “maldição da dimensionalidade” (Harris e Hong, 2001). Este problema surgia já que as regras nebulosas costumavam ser geradas por partições unidimensionais, isto é, cada variável de entrada era submetida a uma partição do seu espaço. Logo, as partições de cada variável de entrada eram combinadas, gerando um número alto de regras nebulosas, e um crescimento exponencial da base de regras a medida que o número de entradas ao sistema crescia. Dependendo de como os algoritmos de agrupamento são utilizados para a geração das regras nebulosas, a maior parte destas abordagens não sofrem mais com o problema da maldição da dimensionalidade, pois a partição não é mais feita para cada dimensão do espaço de entrada de forma independente e sim, para todas de maneira simultânea. Ou seja, a base de regras não é mais uma função exponencial do número de entradas ao sistema computacional.

Dentre as abordagens que consideram a partição do espaço de entrada como um problema multidimensional, destacam-se os algoritmos de agrupamento de dados, como por exemplo, o algoritmo FCM ou *Fuzzy C-Means* (Bezdek, 1981), o algoritmo da montanha ou *Mountain Clustering* (Yager e Filev, 1994), o algoritmo de agrupamento SC ou *Subtractive Clustering* (Chiu, 1994), entre outros. Estes algoritmos têm por objetivo dividir o espaço de entrada em M sub-regiões multidimensionais, baseado no agrupamento de dados similares, onde cada grupo é representado por um centro. Os centros obtidos são utilizados para o cálculo de graus de pertinência associados a cada sub-região nebulosa do espaço de entrada. Estes algoritmos de agrupamento são amplamente aplicados, não só para problemas de classificação de padrões, mas também em problemas de identificação e modelagem de sistemas.

O trabalho descrito em (Harris e Hong, 2001), assim como a proposta de inferência nebulosa utilizada em (Luna et al., 2003), são alternativas que, assim como os algoritmos de agrupamento mencionados, contornam o problema da maldição da dimensionalidade, ou pelo menos, reduzem o processamento para o cálculo da saída do sistema, pois adotam um algoritmo de partição do espaço de entrada adotando uma abordagem multidimensional, ou consideram apenas as regras nebulosas com maior grau de ativação a cada instante de tempo.

Além dos algoritmos de agrupamento, existe uma abordagem baseada na alocação de modelos locais ou especialistas, que está sendo bastante utilizada, principalmente em sistemas adaptativos e de treinamento *online*, como por exemplo, o trabalho proposto em (Murray-Smith e Johansen, 1997) e os descritos em (Angelov e Filev, 2004) e em (Leng et al., 2005).

Abordagens baseadas em modelos locais estão diretamente vinculadas ao princípio de dividir e conquistar, pois o problema é dividido em problemas menores. Os modelos locais são alocados para resolver os sub-problemas, diminuindo assim, a complexidade do problema global a ser resolvido, e facilitando a obtenção de soluções locais de melhor qualidade. Trabalhos como os apresentados em (Murray-Smith e Johansen, 1995), (Murray-Smith e Johansen, 1997) e em (Solomatine e Siek, 2006) são exemplos de aplicações em controle, identificação de sistemas e previsão de fenômenos naturais, respectivamente, sendo todos baseados em sistemas modulares, onde os especialistas são redes neurais ou apenas, modelos de primeira ordem.

A mistura de especialistas é uma outra abordagem que utiliza o princípio de dividir e conquistar. O interesse da comunidade científica em adotar estratégias baseadas em modelos de mistura de especialistas teve um grande aumento entre as décadas de 1980 e 1990, quando surgiu a proposta de mistura de especialistas na área de redes neurais (Jordan, 1986). Inicialmente, o aprendizado de sistemas baseados em mistura de especialistas, conhecido como o algoritmo da Maximização da Verossimilhança ou algoritmo EM (*Expectation Maximization*) foi formulado como um problema de mistura de densidades de probabilidade (Jacobs et al., 1991) a partir da proposta inicial descrita em (Dempster et al., 1977). Posteriormente, foi mostrado que o algoritmo EM poderia ser empregado como uma técnica de aprendizado supervisionado, sendo a sua aplicação estendida a outros campos, tais como o aprendizado de redes neurais multicamadas (Ma et al., 1997), de redes neurais do tipo RBF (Lázaro et al., 2003), assim como diversas aplicações em controle, modelagem (Hong e Harris, 2002) e séries temporais (Carvalho e Tanner, 2005).

O algoritmo EM, como o próprio nome indica (*Expectation Maximization*), consiste em maximizar o valor esperado da função objetivo, definida como a função de verossimilhança do modelo. A função de verossimilhança é baseada na interpretação do modelo como uma mistura de especialistas, com funções de densidade condicional ponderando a mistura dos componentes (para os especialistas) (McLachlan e Basford, 1988), sendo a rede *gating* a responsável pela divisão do problema em sub-problemas.

Embora, a maior parte dos modelos neurais e nebulosos propostos na literatura sejam ajustados via o algoritmo de retropropagação (Haykin, 2001b), o algoritmo EM oferece algumas vantagens quando comparado ao algoritmo tradicional de retropropagação do gradiente

descendente, dentre as quais, destacam-se a rapidez do aprendizado, já que a convergência é atingida com menor número de iterações, assim como a obtenção de ótimos locais melhores. Além disso, mediante a utilização do algoritmo EM para o ajuste de sistemas modulares, e como é o caso deste trabalho, para o ajuste de sistemas computacionais baseados em regras, a interpretação do modelo como um modelo estatístico é mais perceptível. Ou seja, existe uma dualidade inerente ao modelo (Warner e Misra, 1996), tanto para as redes neurais, como para os sistemas nebulosos e os modelos estatísticos.

No entanto, uma desvantagem das técnicas mencionadas para definir o tamanho de sistemas computacionais, baseados em regras ou especialistas, é a necessidade de definir *a priori* o número de partições, e conseqüentemente, o número de regras. No caso de abordagens baseadas em modelos locais ou especialistas, este problema implica o conhecimento prévio da quantidade certa de modelos a ajustar, ou de especialistas necessários para o problema.

Devido à própria complexidade dos problemas reais, os quais são, em geral, de natureza não linear, usualmente não se sabe com antecedência o número adequado de regras ou modelos a serem utilizados. Este problema é normalmente tratado via tentativa e erro. Estruturas de diferentes tamanhos são construídas, ajustadas e testadas. O modelo com melhor desempenho é escolhido como o mais adequado. Este problema é continuamente estudado e, como conseqüência, abordagens de aprendizado adaptativas têm sido desenvolvidas, fornecendo estratégias de aprendizado com alta capacidade de adaptação a baixo custo computacional. Assim, técnicas capazes de determinar a estrutura adequada para cada problema de maneira automática e, de preferência, de maneira não supervisionada, são abordagens desejáveis e promissoras. Nesta direção é que as técnicas de treinamento de redes neurais e sistemas nebulosos têm se desenvolvido nos últimos anos. Os trabalhos desenvolvidos em (Ramamurti e Ghosh, 1996), (Sato e Ishii, 1999), (Er e Wu, 2002), (Angelov e Filev, 2002), (Mao et al., 2005) e em (Leng et al., 2005), (Rong et al., 2006), são apenas alguns exemplos deste tipo de treinamento adaptativo.

Existem duas técnicas gerais de treinamento adaptativo. A abordagem construtiva *offline* consiste em um algoritmo de aprendizado em que o total de dados de treinamento é apresentado a cada iteração ou época. A medida que o aprendizado vai se desenvolvendo, a estrutura do modelo é modificada por meio de mecanismos de geração e incorporação de novas unidades na estrutura, assim como de critérios de poda ou eliminação.

Na abordagem de aprendizado adaptativo *online*, também chamado aprendizado seqüencial, considera-se a estrutura do sistema variante à medida que os dados vão sendo processados, ou seja, a estrutura do modelo é modificada de maneira dinâmica, dependendo das

necessidades do entorno. A cada instante de tempo, o espaço de entrada é dividido de acordo com a necessidade de um novo modelo local. Toda vez que o desempenho do sistema é abaixo do desejado, considera-se a possibilidade de incorporar um novo modelo local ou especialista à estrutura do sistema. Quando isto acontece, conseqüentemente, uma nova partição do espaço de entrada é gerada, re-posicionando as sub-regiões já existentes e modificando assim, a arquitetura do modelo computacional. Técnicas de poda ou eliminação de módulos também são aplicadas neste tipo de aprendizado, já que, mediante a aplicação de operadores de adição (*adding*) e eliminação (*pruning*) de maneira alternada durante o treinamento do sistema computacional, obtém-se modelos eficientes e de estrutura mais compacta (Ramamurti e Ghosh, 1996). No caso de modelos com aprendizado adaptativo *online*, a aplicação de operadores de adição e de poda é permanente.

Os modelos locais utilizados, como já foi mencionado, podem ser lineares ou não lineares, dependendo da complexidade do problema e das exigências da modelagem, tanto em termos de erro, como em facilidade de adaptação e/ou complexidade da própria estrutura do sistema.

Em problemas de alta complexidade, como é o caso do planejamento energético no Brasil, onde mais de 90 séries históricas de vazões são modeladas, é necessário adotar estratégias simples, mas eficientes, que facilitem a modelagem e que sejam adaptáveis a mudanças na dinâmica do problema a longo do tempo. As técnicas atualmente empregadas consideram a modelagem de cada uma das séries utilizando um total de doze modelos de previsão, ou seja, um modelo para cada mês do histórico de vazões, associado a cada um dos postos de aproveitamento, localizados em diferentes regiões do Brasil. No caso das séries de vazões semanais, o número de modelos a serem ajustados para cada série é igual a 52, ou seja, um modelo para cada semana (CEPEL, 1997). Neste contexto, abordagens baseadas em modelos locais, ou no princípio de dividir e conquistar, mas com ajuste “paralelo” e com a determinação e atualização da estrutura de forma automática são de grande interesse.

As pesquisas em modelos híbridos, como são os modelos neurais nebulosos, também têm se voltado para estas abordagens de aprendizado. Uma revisão geral sobre as diferentes estruturas de modelos neurais-nebulosos e os seus métodos de aprendizado são descritos em (Mitra e Hayashi, 2000).

Os sistemas nebulosos baseados no modelo de Takagi-Sugeno têm se convertido em uma poderosa ferramenta na área de engenharia para modelagem e controle de sistemas complexos (Angelov e Filev, 2004), pois utilizam uma combinação não linear de modelos simples, localizados em sub-regiões do espaço de entrada. Dado que estes sistemas têm mostrado serem eficientes para diversos problemas, incluindo séries temporais, este trabalho propõe um

sistema híbrido, o qual é composto por uma base de regras nebulosas do tipo se-então, sendo o algoritmo de aprendizado baseado no algoritmo EM. A vantagem de adotar técnicas de aprendizado baseadas no algoritmo EM, é a convergência rápida do algoritmo, assim como a obtenção de ótimos locais melhores, quando comparado ao algoritmo de retropropagação temporal (Haykin, 2001b).

Assim, a proposta desta tese consiste na construção de modelos de previsão a partir de sistemas computacionais baseados em regras nebulosas, sendo a estrutura e os parâmetros do modelo ajustados utilizando técnicas inspiradas em modelos locais e de redes neurais.

A etapa de aprendizado proposta nesta tese é realizada de duas maneiras: primeiro, adotando um mecanismo de aprendizado *offline* construtivo, e depois, adotando um mecanismo de aprendizado *online* e adaptativo. Ambas as técnicas de aprendizado propostas definem de maneira automática a estrutura do sistema, e ambos os algoritmos - *offline* e *online* - são baseados no algoritmo EM. Para o caso do aprendizado *online*, foi proposto o algoritmo de aprendizado EM recursivo, de tal forma que não seja necessário ter conhecimento do histórico completo a cada iteração, e sim, apenas do dado atual.

Os trabalhos descritos em (Ramamurti e Ghosh, 1996) e em (Sato, 2000) são trabalhos similares à proposta de aprendizado construtivo e *online*, respectivamente. No entanto, aquelas abordagens foram apenas utilizadas para o ajuste de mistura de especialistas e, no caso de Sato (2000), para modificações em técnicas de aprendizado por reforço (Sato e Ishii, 1999).

Já no caso desta tese, as propostas de aprendizado apresentam uma série de características distintas em relação às propostas mencionadas na literatura, tais como o processo de inicialização e o próprio processo de adaptação, assim como o enfoque do próprio modelo computacional, como será visto nos próximos capítulos.

3.2 Modelos comparativos

Esta seção descreve brevemente alguns dos modelos que serão utilizados para comparação e avaliação do desempenho dos modelos propostos neste trabalho.

Os modelos não lineares da área de inteligência computacional utilizados neste trabalho são a rede neural multicamadas **MLP** (Haykin, 1994), a rede neural nebulosa **ANFIS** (Jang, 1993), a rede neural nebulosa **RNN** (Figueiredo, 1997) e o sistema evolutivo baseado em regras nebulosas **eTS** (Angelov e Filev, 2005).

Embora exista uma grande quantidade de arquiteturas de redes neurais, sem dúvida a rede

multicamada (**MLP** - *Multilayer Perceptron*) é a mais encontrada e empregada na literatura (Hush e Horne, 1993), devido a sua capacidade de aproximação universal e flexibilidade para formar soluções de qualidade para uma ampla classe de problemas, a partir de um mesmo algoritmo de aprendizado (Iyoda, 2000). Por esse motivo a rede MLP será utilizada para fins de análise e comparação de resultados.

A estrutura de rede neural multicamadas utilizada neste trabalho será uma rede neural composta por uma camada de entrada, uma camada oculta e uma camada de saída. Os neurônios da camada oculta possuem funções de ativação do tipo sigmóide, enquanto os neurônios das camadas de entrada e de saída possuem funções de ativação lineares. O número de neurônios da camada oculta varia para cada aplicação e o algoritmo utilizado para o ajuste da rede neural é o algoritmo tradicional de retropropagação do erro, com taxas de aprendizado variantes a cada época.

Dentre os modelos híbridos baseados em regras nebulosas, destaca-se o sistema de inferência neural nebuloso adaptativo ou **ANFIS** (Jang, 1993). Dado um conjunto de padrões entrada-saída, o **ANFIS** constrói um sistema de inferência neural nebuloso, sendo os parâmetros associados com as funções de pertinência ajustados via o algoritmo de retropropagação. Neste trabalho, o modelo **ANFIS** foi construído utilizando o toolbox do Matlab® (*Fuzzy Logic Toolbox, For Use with MATLAB®*, 2000), considerando regras nebulosas com conseqüentes lineares dependentes das variáveis de entrada, e antecedentes inicializados com centros uniformemente distribuídos no espaço entrada-saída. As funções de pertinência são representadas por funções gaussianas. Tanto os parâmetros dos antecedentes como dos conseqüentes das regras nebulosas são otimizados durante o aprendizado do sistema de inferência.

Um outro modelo híbrido que será empregado neste trabalho é a rede neural nebulosa (**RNN**) proposto em (Figueiredo, 1997) e aplicada à série de vazões em (Ballini, 2000). A **RNN** tem uma estrutura adaptativa sendo o método de aprendizado do tipo construtivo, onde regras nebulosas são adicionadas à estrutura da rede quando o desempenho não é satisfatório. Neste trabalho, o modelo RNN foi construído utilizando o sistema HydroPrev, o qual constitui uma categoria de modelos de previsão de vazões dentro do programa HydroLab (Cicogna, 2003).

O conceito de regras nebulosas evolutivas foi proposto por Angelov e Filev (2002), sendo recentemente sugeridas algumas modificações (Angelov e Filev, 2004), (Angelov e Filev, 2005). Já que este trabalho propõe um modelo baseado em regras nebulosas com aprendizado seqüencial, é necessário validar os resultados obtidos e compará-los a abordagens similares. Neste sentido, os modelos propostos serão comparados ao modelo evolutivo **eTS**, o qual, diferente

das propostas deste trabalho não possui uma etapa de inicialização da estrutura da base de regras nebulosas.

Originalmente, o algoritmo de aprendizado **eTS** foi baseado na atualização recursiva dos parâmetros associados à base de regras, atualizando de forma dinâmica o número de regras e os respectivos parâmetros dos antecedentes destas regras. Esse processo é combinado com um ajuste recursivo de quadrados mínimos para a atualização dos parâmetros dos conseqüentes.

Uma versão computacional e simplificada do algoritmo **eTS** foi recentemente sugerida por Angelov e Filev (2005), com a finalidade de reduzir a complexidade do algoritmo original, sendo o algoritmo chamado de **simpleTS**. No caso da proposta apresentada nesta tese, o algoritmo de aprendizado recursivo é baseado no algoritmo EM e considera durante o processo de adaptação, critérios de geração de novas regras e de poda ou eliminação de regras nebulosas de forma simultânea, sendo esta característica um diferencial da proposta desta tese, quando comparado com os outros modelos mencionados. O ajuste dos modelos **eTS** e **simpleTS** foram realizados utilizando a implementação dos mesmos disponíveis em (Angelov e Zhou, 2006).

Por outro lado, devido as aplicações que serão apresentadas no Capítulo 7, é necessário comparar os modelos propostos com modelos tradicionais da área de séries temporais. Desta maneira, os modelos **ARIMA**, **MSA**, **TAR** e **PAR** serão utilizados como base de comparação para algumas das aplicações apresentadas no Capítulo 7, a partir dos resultados descritos em (Tsay, 2005). Maiores detalhes referentes a estes modelos de série temporais podem ser encontrados em (Morettin e Toloi, 2004) e também em (Tsay, 2005).

3.3 Resumo

Este capítulo apresentou uma breve revisão bibliográfica de modelos não lineares de séries temporais, com enfoque principalmente, em modelos baseados em regras nebulosas.

Além disso, este capítulo apresentou uma breve descrição dos modelos que serão ajustados com o objetivo de analisar o desempenho dos modelos propostos neste trabalho.

4 *Estrutura de modelo de série temporal*

4.1 Introdução

A estrutura do modelo de série temporal proposto neste trabalho é composto por um conjunto de regras nebulosas do tipo Takagi-Sugeno, o qual também pode ser interpretado como um conjunto de modelos locais.

A proposta deste trabalho, assim como algumas outras propostas da literatura, como é o caso dos modelos evolutivos propostos por Kasabov e Song (2002) e por Angelov e Filev (2004) utilizam a idéia de decompor o problema não-linear por meio da partição do problema em sub-problemas, sob sub-regiões nebulosas do espaço de entrada.

Estes sub-problemas são combinados de tal forma que a saída do modelo é estimada como uma combinação não-linear de modelos locais simples que, neste trabalho, são definidos como modelos locais lineares ou de regressão. Esta característica torna o modelo proposto interessante pois, mediante a geração de soluções locais simples, o modelo fornece uma solução para o problema global cuja complexidade é superior à dos sub-problemas definidos pela partição nebulosa do espaço de entrada, a qual é gerada de forma automática e simultânea.

4.2 Estrutura geral do modelo

O sistema adaptativo proposto é baseado em regras nebulosas e principalmente, em modelos de primeira ordem, compondo um sistema nebuloso do tipo Takagi-Sugeno (TS) (Takagi e Sugeno, 1985). Assim, o sistema tem um mecanismo de inferência nebulosa, codificando na sua estrutura uma base de regras do tipo **Se-Então**, adotando a seguinte representação:

$$\mathbf{Se} < \text{antecedente} > \mathbf{Ent\tilde{a}o} < \text{consequente} > \quad (4.1)$$

A parte do antecedente representa uma sub-região do espaço de entrada. Esta sub-região define a região ativa associada à regra nebulosa. A parte do consequente é definida por um modelo local que contribuirá com uma parcela na estimativa da saída final.

A estrutura geral de um modelo composto por M regras nebulosas é ilustrada na Figura 4, onde $\mathbf{x}^k = [x_1^k, x_2^k, \dots, x_p^k]$ é o vetor de entrada no instante k , $k \in \mathbb{Z}_0^+$; \hat{y}^k é a saída do modelo, para uma determinada entrada \mathbf{x}^k .

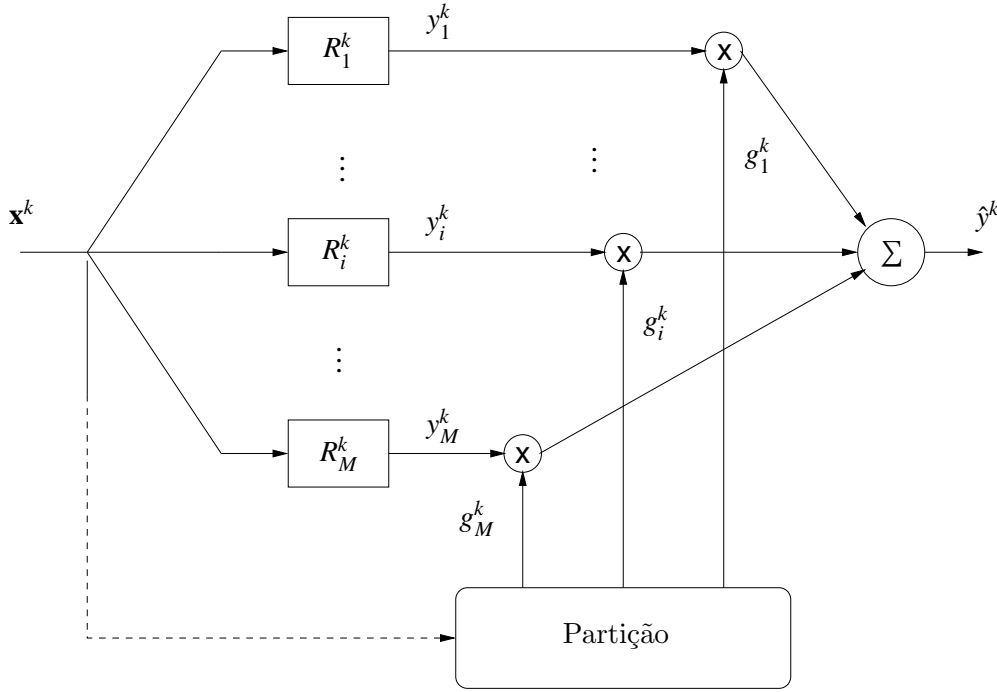


Figura 4: Estrutura do modelo proposto composto por M regras nebulosas.

O grau de pertinência de cada padrão de entrada \mathbf{x}^k é representado por $g_i(\mathbf{x}^k) = g_i^k \in [0, 1]$, o qual é calculado da seguinte forma:

$$g_i(\mathbf{x}^k) = g_i^k = \frac{\alpha_i \cdot P[i | \mathbf{x}^k]}{\sum_{q=1}^M \alpha_q \cdot P[q | \mathbf{x}^k]} \quad (4.2)$$

sendo α_i coeficientes positivos que satisfazem a condição:

$$\sum_{i=1}^M \alpha_i = 1 \quad (4.3)$$

Estes coeficientes também podem ser considerados como índices indiretos de relevância para cada regra gerada durante o treinamento, de tal forma que, quanto mais alto seja o valor de α_i , para $i = 1, \dots, M$, maior será a importância da i -ésima regra nebulosa.

Em (4.2), $P[i | \mathbf{x}^k]$ é a probabilidade condicional de i dado \mathbf{x}^k :

$$P[i | \mathbf{x}^k] = \frac{1}{(2\pi)^{p/2} \det(\mathbf{V}_i)^{1/2}} \exp \left\{ -0,5 \times (\mathbf{x}^k - \mathbf{c}_i) \mathbf{V}_i^{-1} (\mathbf{x}^k - \mathbf{c}_i)^T \right\} \quad (4.4)$$

na qual \mathbf{V}_i é uma matriz de ordem $p \times p$ e representa a covariância associada à probabilidade $P[i | \mathbf{x}^k]$.

Observa-se que $P[i | \mathbf{x}^k]$ é uma função de densidade de probabilidade condicional, com centros em \mathbf{c}_i e matriz de covariância \mathbf{V}_i , a qual representa a probabilidade de escolher a regra i , dado que apenas a entrada \mathbf{x}^k é conhecida, assim como os parâmetros \mathbf{V}_i e \mathbf{c}_i .

A Equação (4.4) também pode ser interpretada como a probabilidade do vetor de entrada \mathbf{x}^k pertencer à região ativa da i -ésima regra, já que, a medida que o dado de entrada encontra-se mais próximo do i -ésimo centro, maior será o valor do i -ésimo grau de pertinência. Em outras palavras, cada sub-região nebulosa é representada pelo respectivo centro \mathbf{c}_i , assim como os parâmetros de dispersão codificados na diagonal principal da matriz de covariância associada \mathbf{V}_i .

Além disso, da Equação (4.2), observa-se que g_i^k satisfaz a seguinte condição:

$$\sum_{i=1}^M g_i^k = 1 \quad (4.5)$$

sendo esta condição necessária para considerar g_i^k como uma função de pertinência (Pedrycz e Gomide, 1998).

A saída do sistema proposto, \hat{y}^k , é calculada como uma combinação não linear de modelos locais com saídas y_i^k e os seus respectivos graus de pertinência g_i^k , ou seja:

$$\hat{y}^k = \sum_{i=1}^M g_i^k y_i^k \quad (4.6)$$

onde, neste caso, y_i , para $i = 1, \dots, M$, é definido por:

$$y_i^k = \phi^k \times \theta_i^T \quad (4.7)$$

sendo $\phi^k = [1 \ x_1 \ x_2 \ \dots \ x_p]$ o vetor composto por um termo constante mais as variáveis de entrada; $\theta_i = [\theta_{i0} \ \theta_{i1} \ \dots \ \theta_{ip}]$ o vetor de coeficientes dos modelos locais lineares, de tamanho $1 \times (p+1)$. Nas Equações (4.2)-(4.6), foi considerada apenas uma saída, no entanto, o modelo pode ser estendido para múltiplas saídas.

A dinâmica do modelo pode ser interpretada como um mecanismo de inferência nebulosa, sendo a base de regras composta por regras R_i^k , $i = 1, \dots, M$, $k = 1, \dots, N$, definidas como segue:

$$\begin{aligned} R_i^k : \quad & \text{Se } < \mathbf{x}^k \text{ pertence à região "i" com um grau de pertinência } g_i^k > \\ & \text{Então } < y_i^k = \phi^k \times \theta_i^T > \end{aligned} \quad (4.8)$$

para $i = 1, \dots, M$.

Por outro lado, o modelo proposto pode também ser interpretado como uma arquitetura modular, sendo que o mecanismo de inferência representa o sistema de controle para cada módulo ou especialista (Guo e Xu, 1999), e o módulo em si é representado por cada um dos modelos lineares definidos.

Ou seja, existe uma analogia entre o modelo de mistura de especialistas e o sistema computacional proposto, já que a saída do sistema é calculada como o valor ponderado das saídas de cada regra nebulosa. Os coeficientes que ponderam as saídas locais, da mesma forma que o *gating* para modelos de mistura de especialistas, também dependem das entradas a cada instante e, dado que as funções de pertinência são definidas como funções gaussianas generalizadas, estes podem simultaneamente, representar funções de probabilidade e fazer uso do algoritmo EM para ajustar o modelo proposto.

Um trabalho similar é desenvolvido por Harris e Hong (2001), onde considera-se outros aspectos na modelagem devido ao problema da maldição da dimensionalidade, como por exemplo o processamento de um sub-conjunto de regras nebulosas ao invés do total de regras que compõem o modelo neural-nebuloso, a independência dos modelos locais embutidos na estrutura do modelo e o processamento paralelo. O cálculo da saída do sistema nebuloso considerando apenas um sob-conjunto de regras nebulosas também foi apontado em (Luna,

2003).

Neste capítulo, a estrutura descrita e utilizada como parte da proposta desta tese, é apenas uma arquitetura geral, que será otimizada de acordo com o algoritmo de treinamento adotado, de forma que não seja necessário definir previamente o número de regras M necessárias na estrutura do modelo baseado em regras nebulosas.

A vantagem da estrutura proposta é a sua relativa simplicidade quando comparada a outras abordagens não lineares, assim como a sua capacidade de interpretação e adaptação. Além disso, esta permite uma série de variações no treinamento do modelo, obtendo ganhos tais como, a seleção automática da estrutura, um processo de aprendizado mais rápido, capacidade de adaptação, entre outras.

Diferentes trabalhos da literatura têm levantado comparações entre modelos puramente estatísticos e modelos da inteligência computacional, como é o caso por exemplo, dos trabalhos descritos por Warner e Misra (1996), Kihoro et al. (2004) e por Tsay (2005).

Ambas as abordagens são, em geral, utilizadas com o intuito de modelar dependências entre as variáveis envolvidas, sendo o modelo freqüentemente representado como segue:

$$\hat{y}^k = f[\mathbf{x}^k] = f[x_1^k, x_2^k, \dots, x_p^k]$$

onde $f[\cdot]$ é uma relação funcional entre a variável independente ou observável \mathbf{x}^k e a variável dependente y^k ; \hat{y}^k denota o valor estimado de y^k .

Desta forma, o objetivo dos modelos de regressão, assim como dos modelos propostos neste trabalho, consiste na predição de uma variável dependente a partir de variáveis de entrada observáveis \mathbf{x}^k , comumente, mas não sempre, composta por atrasos das realizações da saída. Este valor previsto \hat{y}^k , é representado por uma relação não-linear definida pela Equação (4.6).

A principal diferença entre ambas as abordagens é que, o modelo de regressão é definido por um vetor de coeficientes enquanto os modelos baseados em redes neurais ou sistemas nebulosos, como os propostos neste trabalho, são definidos por um conjunto de parâmetros, incluindo centros \mathbf{c}_i , matrizes de covariância \mathbf{V}_i e os parâmetros dos modelos locais que representam os conseqüentes das regras nebulosas θ_i , $i = 1, \dots, M$.

Portanto, enquanto os modelos lineares possuem uma forma fechada definida pelos seus coeficientes, os modelos com aprendizado construtivo ou *online* re-definem os seus parâmetros durante um processo de ajuste ou aprendizado. Apesar deste “custo extra”, a vantagem

de adotar modelos não-lineares com algoritmos de aprendizado como os propostos, é que aplicando estes algoritmos, tanto a estrutura do modelo como o conjunto de parâmetros são definidos de forma automática, diminuindo a dificuldade na escolha dos parâmetros e de uma estrutura adequada para a construção dos modelos de séries temporais.

4.3 Resumo

Este capítulo apresentou uma descrição geral da estrutura do modelo baseado em regras nebulosas que será utilizada nos próximos capítulos com duas técnicas de aprendizado diferentes. O modelo descrito representa um mecanismo de inferência nebulosa com regras do tipo **Se-Então**.

Devido à representação linear dos conseqüentes das regras que compõem a estrutura do modelo, este torna-se relativamente simples já que o problema é decomposto em sub-problemas de menor complexidade, onde as soluções locais são dadas pelos conseqüentes das regras nebulosas, gerando desta forma, uma solução global do problema não-linear em estudo, de acordo com a idéia de “dividir para conquistar”.

5 *Modelo baseado em regras nebulosas com aprendizado construtivo offline*

5.1 Introdução

Os modelos baseados em regras nebulosas têm-se mostrado promissores, principalmente quando aplicados em problemas de identificação e controle de sistemas. Alguns trabalhos recentes da literatura, como por exemplo, as propostas apresentadas em (Nie, 1997), (Kasabov e Song, 2002) e em (Mao et al., 2005) têm desenvolvido trabalhos aplicando este tipo de modelos para a previsão de série temporais.

A principal vantagem dos modelos baseados em regras nebulosas é a sua simplicidade em termos de estrutura gerada, sendo possível em alguns casos, obter uma representação lingüística que fornece um melhor entendimento do problema tratado.

Este capítulo propõe um modelo baseado em regras nebulosas, com aprendizado construtivo e *offline*, o qual é baseado no algoritmo da maximização da verossimilhança (EM). A principal característica do modelo proposto é a geração automática da estrutura do modelo, a partir de uma inicialização com uma estrutura mínima, o que simplifica a modelagem.

O capítulo finaliza a apresentação da proposta com um exemplo demonstrativo, que envolve a modelagem de uma série temporal, de tal forma que a adaptação da estrutura e de alguns dos parâmetros do modelo seja visível, facilitando assim a compreensão da proposta.

5.2 Modelo construtivo baseado em regras nebulosas (C-FSM)

O modelo construtivo baseado em regras nebulosas (**C-FSM**), é uma abordagem que tem por objetivo definir de forma automática o número de regras necessárias para cada aplicação, por meio de um aprendizado *offline* e construtivo.

O algoritmo de aprendizado é chamado de *offline* já que este é aplicado considerando a cada iteração, todos os dados disponíveis para o seu ajuste, sendo este procedimento também conhecido como aprendizado por lote ou batelada (Haykin, 2001b). Ao mesmo tempo, o algoritmo é chamado de construtivo, devido à adaptação da estrutura do modelo durante o aprendizado, acrescentando ou eliminando regras que compõem a estrutura do modelo.

O processo de aprendizado construtivo proposto neste capítulo pode ser dividido em duas etapas: a etapa de inicialização e a etapa de aprendizado construtivo, no qual, por meio de operadores de adição e punição, a estrutura do modelo é modificada durante uma sequência de iterações, sendo os parâmetros do modelo ajustados de forma simultânea. Este processo continua até satisfazer um critério pré-definido ou uma condição de parada. As etapas que constituem o algoritmo são descritos a continuação.

5.2.1 Primeira etapa: inicialização

Inicialmente, uma estrutura é gerada, tendo como base a estrutura do sistema híbrido descrita no Capítulo 4. Esta estrutura inicial é composta por um número reduzido de regras nebulosas do tipo Takagi-Sugeno.

Os parâmetros do modelo inicial são ajustados utilizando o algoritmo EM, proposto em (Jacobs et al., 1991), (Jordan e Jacobs, 1993). Com base neste algoritmo de otimização, os parâmetros do modelo são ajustados a partir de um conjunto de dados de treinamento durante uma sequência iterativa de passos E (de *Expectation*) e M (de *Maximization*) (Jacobs et al., 1991), sendo a solução ótima para cada passo EM dada por um conjunto de equações que serão apresentadas a seguir, junto com uma breve descrição do algoritmo EM tradicional.

5.2.1.1 O algoritmo EM

O algoritmo da Maximização da Verossimilhança tradicional foi originalmente proposto na área de redes neurais como uma técnica de otimização para mistura de especialistas em

(Jacobs et al., 1991). Baseado no algoritmo EM, os parâmetros do modelo são ajustados utilizando um conjunto de N dados entrada-saída, durante uma sequência iterativa de passos EM. O objetivo deste algoritmo é encontrar um conjunto de parâmetros para o modelo, de tal forma que a função objetivo seja maximizada. Esta função objetivo é tradicionalmente definida como o logaritmo da função de verossimilhança \mathcal{L} , dos valores observados de y^k a cada passo M do processo de aprendizado. Esta função objetivo é definida por:

$$\mathcal{L}(D, \mathbf{\Omega}) = \sum_{k=1}^N \ln \left(\sum_{i=1}^M g_i(\mathbf{x}^k, \mathbf{C}) \times P(y^k | \mathbf{x}^k, \theta_i) \right) \quad (5.1)$$

na qual $D = \{(\mathbf{x}^k, y^k) | k = 1, \dots, N\}$ é o conjunto de pares entrada-saída, $\mathbf{\Omega}$ é conjunto de todos os parâmetros do modelo e \mathbf{C} contém apenas os parâmetros dos antecedentes (centros e matrizes de covariância). Para maximizar $\mathcal{L}(D, \mathbf{\Omega})$, é necessário estimar uma série de variáveis faltantes ou incompletas h_i^k durante o passo M (Moerland, 1997), (Lázaro et al., 2003). Estas variáveis, de acordo com a teoria de mistura de especialistas, são interpretadas como a probabilidade a posteriori do i -ésimo modelo local ser ativado, dada a entrada \mathbf{x}^k . Ou seja, é uma estimativa da função ou grau de pertinência, dado que, tanto a entrada \mathbf{x}^k como a saída y^k são conhecidas.

Quando o algoritmo EM é adaptado para o ajuste de modelos baseados em regras nebulosas, a variável h_i^k pode ser interpretada como a estimativa a posteriori das funções de pertinência g_i^k , as quais são definidas no Capítulo 4, pela Equação (4.2), para $i = 1, \dots, M$. Estas estimativas são calculadas de acordo com a seguinte Equação:

$$h_i^k = \frac{\alpha_i P(i | \mathbf{x}^k) P(y^k | \mathbf{x}^k, \theta_i)}{\sum_{q=1}^M \alpha_q P(q | \mathbf{x}^k) P(y^k | \mathbf{x}^k, \theta_q)} \quad (5.2)$$

para $i = 1, \dots, M$. Como já foi mencionado, estas estimativas são chamadas de “posteriori”, pois estas são calculadas considerando que \mathbf{x}^k e y^k , $k = 1, \dots, N$, são conhecidas, o que difere da estimativa fornecida pela Equação (4.2). Além disso, a probabilidade condicional $P(y^k | \mathbf{x}^k, \theta_i)$ é definida por:

$$P(y^k | \mathbf{x}^k, \theta_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left(-\frac{[y^k - y_i^k]^2}{2\sigma_i^2} \right) \quad (5.3)$$

sendo a variância σ_i^2 estimada como:

$$\sigma_i^2 = \frac{\sum_{k=1}^N h_i^k (y^k - y_i^k)^2}{\sum_{k=1}^N h_i^k} \quad (5.4)$$

Assim, utilizando o algoritmo EM para o ajuste dos parâmetros do modelo construtivo **C-FSM**, obtém-se uma solução ótima, a qual é definida pelas seguintes equações:

$$\alpha_i = \frac{1}{N} \sum_{k=1}^N h_i^k \quad (5.5)$$

$$\mathbf{c}_i = \frac{\sum_{k=1}^N h_i^k \mathbf{x}^k}{\sum_{k=1}^N h_i^k} \quad (5.6)$$

$$\mathbf{V}_i = \frac{\sum_{k=1}^N h_i^k (\mathbf{x}^k - \mathbf{c}_i)^T \times (\mathbf{x}^k - \mathbf{c}_i)}{\sum_{k=1}^N h_i^k} \quad (5.7)$$

para $i = 1, \dots, M$, onde M é o tamanho da base de regras e N é o número de dados entrada-saída que compõem o conjunto de treinamento para o **C-FSM**. Estas equações foram obtidas considerando \mathbf{V}_i como uma matriz diagonal e definida positiva, pois desta forma evita-se o problema de soluções infactíveis.

Uma solução ótima para θ_i surge a partir da solução de:

$$\sum_{k=1}^N \frac{h_i^k}{\sigma_i^2} (y^k - \phi^k \times \theta_i) \cdot \phi^k = 0 \quad (5.8)$$

na qual σ_i^2 é a variância associada a cada uma das saídas locais y_i^k , $i = 1, \dots, M$ ou conseqüentes das regras, $\phi^k = [1 \ \mathbf{x}^k]$ e θ_i é o vetor dos coeficientes dos modelos locais.

O algoritmo EM tradicional é resumido nos seguintes passos:

Algoritmo 4 EM tradicional

Passo 1. Passo E: Estimar h_i^k via Eq. (5.2);

Passo 2. Passo M: Maximizar Eq. (5.1) e atualizar os parâmetros do modelo, cujos valores ótimos a cada iteração são dados pelas Equações (5.5)-(5.8);

Passo 3. Após o ajuste, calcular o novo valor para a função objetivo $\mathcal{L}(D, \mathbf{\Omega})$. Se a convergência ou condição de parada é atingida, então parar o processo, senão, voltar ao Passo 1.

O algoritmo EM tradicional será utilizado nesta fase de inicialização, para uma estrutura contendo duas regras nebulosas, sendo os parâmetros da base de regras inicializados da seguinte maneira:

- O vetor de centros \mathbf{c}_i é inicializado de tal forma que seja igual a um dado de entrada do conjunto de treinamento, o qual é escolhido de forma aleatória;
- $\alpha_i = 1/M = 0,5$;
- $\mathbf{V}_i = 10^{-4}\mathbf{I}$, onde \mathbf{I} é uma matriz identidade;
- Os coeficientes θ_i são inicializados com valores aleatórios no intervalo $[0, 1]$;
- O desvio padrão $\sigma_i = 1,0$, o qual é associado à i -ésima regra nebulosa.

para $i = 1, 2$. As soluções ótimas locais definidas para cada iteração do algoritmo EM são importantes não somente para a inicialização do modelo, mas também para a etapa de adaptação da estrutura e parâmetros do modelo, já que estas soluções serão utilizadas durante o aprendizado construtivo do modelo **C-FSM**, o qual é descrito a seguir.

5.2.2 Segunda etapa: treinamento e adaptação

O algoritmo construtivo consiste na modificação da estrutura gerada na etapa de inicialização. Estas modificações são executadas durante o processo de aprendizado, mediante a aplicação de operadores de adição e de eliminação de regras nebulosas em conjunto com o algoritmo EM. Previamente ao aprendizado da estrutura, é necessário definir critérios de adição e de poda, baseados em condições desejadas ou pré-definidas. Quando o modelo atual não satisfaz as condições desejadas, então a estrutura do modelo é modificada, seja gerando novas regras ou eliminando regras existentes como parte da estrutura do modelo. No caso

em que uma das regras que compõem a estrutura atual do modelo não é mais relevante, esta é eliminada da estrutura. Assim, o novo modelo é atualizado e re-ajustado aplicando uma sequência de passos EM, tendo sempre em consideração as restrições e condições definidas para os parâmetros através das Equações (5.5)-(5.8).

Os critérios para a aplicação dos operadores de adição e de poda de regras nebulosas são descritos a seguir.

5.2.2.1 Geração de uma nova regra o adição

O critério definido para a geração de uma nova regra nebulosa é chamado critério do antecedente, pois este tem como objetivo alocar os dados de entrada em regiões ativas definidas para as regras nebulosas. Dado que cada uma das M partições é representada pelo antecedente de cada uma das M regras nebulosas, o critério é a princípio orientado à geração ótima dos antecedentes destas regras.

O critério do antecedente avalia a capacidade de cada regra nebulosa *gerar* cada um dos dados de entrada e assim, cobrir de forma adequada o espaço de entrada definido pelos dados apresentados. Isto é, cada dado de entrada deve ser *gerado* por uma das regras locais com uma probabilidade mínima superior a um limiar pré-definido. Isto implica na localização do dado de entrada na região ativa da regra nebulosa.

Seja $R_{i^*}^k$ a regra que fornece o maior grau de pertinência para \mathbf{x}^k , $i^* \in \{1, 2, \dots, M\}$, $k = 1, 2, \dots, N$. Considerando que os dados seguem uma distribuição normal e dado o nível de confiança de $\gamma\%$, pode-se construir o intervalo de confiança $[\mathbf{c}_i - z_\gamma \sqrt{\text{diag}(\mathbf{V}_i)}, \mathbf{c}_i + z_\gamma \sqrt{\text{diag}(\mathbf{V}_i)}]$, no qual \mathbf{c}_i é o centro e $\text{diag}(\mathbf{V}_i)$ é a diagonal principal da matriz \mathbf{V}_i . Neste trabalho, foi considerado um nível de confiança $\gamma = 72,86\%$, tal que $P(-z_\gamma < Z < z_\gamma) = 72,86\%$, com $Z \sim N(0, 1)$, como ilustra a Figura 5.

Assim, a probabilidade de \mathbf{x}^k ativar a i -ésima regra (ver Equação (4.4)) deve ser superior a 0,1357, de acordo com a tabela de distribuição normal. Ou seja:

$$P[i^* | \mathbf{x}^k] > 0,1357 \quad (5.9)$$

com:

$$i^* = \arg \max_i \left(P[i | \mathbf{x}^k] \right)_{i=1, \dots, M} \quad (5.10)$$

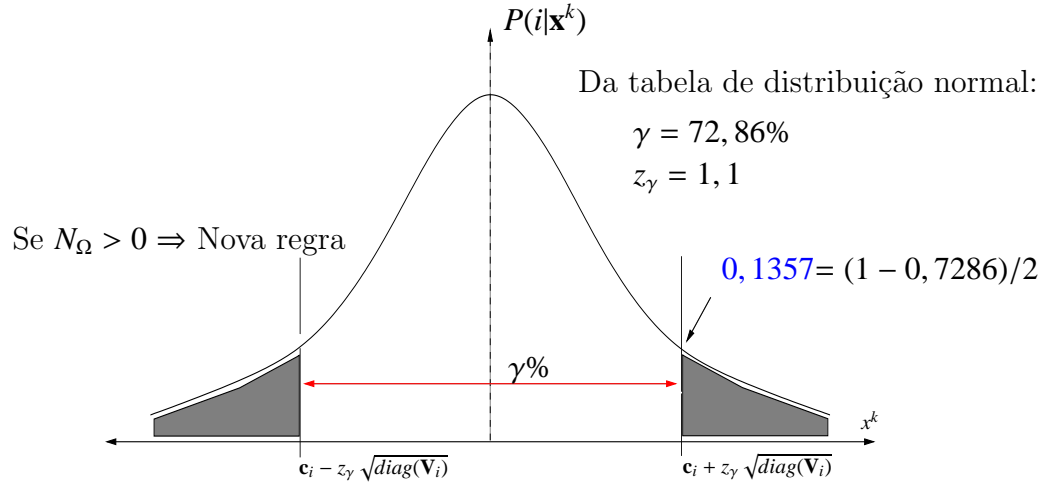


Figura 5: Condição para gerar uma nova regra nebulosa.

Se a condição definida pela Equação (5.9) não for satisfeita por um ou mais dados do conjunto de treinamento, significa que estes dados possuem uma probabilidade de pertencer a alguma região ativa coberta pela base de regras, inferior ao desejado. Denotando por Ω o conjunto contendo todos os dados entrada-saída que não satisfazem a Equação (5.9), uma nova regra será inserida à estrutura do sistema, ou seja:

$$M = M + 1 \quad (5.11)$$

A nova regra com região de ativação tem por objetivo incluir os dados ainda não considerados pelas regras já existentes, de tal forma que a partição do espaço de entrada seja otimizada, obtendo assim um novo ótimo local por meio do re-ajuste do modelo. Desta maneira, o centro da nova regra é inicialmente estimada como:

$$\mathbf{c}_{M+1} = \frac{1}{N_{ng}} \sum_{t \in \Omega} \mathbf{x}(t) \quad (5.12)$$

sendo N_{Ω} o número de dados $[\mathbf{x}^t \ y^t]$ em Ω . Além disso, os demais parâmetros da nova regra nebulosa são inicializados como segue:

- $\sigma_{M+1} = 1, 0$;
- $\theta_{M+1} = [\bar{y} \ 0 \ \dots \ 0]$, onde \bar{y} é calculado como o valor médio das saídas desejadas pertencentes ao conjunto Ω .

Os parâmetros α_i e \mathbf{V}_i , para $i = 1, \dots, M$, devem ser re-inicializados de acordo com:

- $\mathbf{V}_i = 10^{-4}\mathbf{I}$, onde $\mathbf{I}_{p \times p}$ é a matriz identidade;
- $\alpha_i = 1/(M)$.

Após a inserção da nova regra, a estrutura do sistema é re-ajustada durante algumas iterações EM. Como α_i indica a relevância de cada regra nebulosa, a sua re-inicialização é necessária para que todas as regras tenham o mesmo peso, fazendo com que o ajuste dos parâmetros seja não-tendencioso, obtendo assim, uma nova partição do espaço de entrada. A re-inicialização de \mathbf{V}_i é necessária já que o processo de re-ajuste definirá novos valores de dispersão para cada regra nebulosa, de acordo com a nova partição gerada, ou seja, considerando neste processo a nova regra inserida na estrutura do modelo.

5.2.2.2 Eliminação ou poda

Os parâmetros α_i , $i = 1, \dots, M$ podem ser considerados como uma medida indireta da importância ou relevância de cada regra nebulosa para a arquitetura atual do modelo de série temporal, com relação as outras regras, isto devido à condição de unicidade definida no Capítulo 4, Equação (4.3), e a definição de α_i dada pela Equação (5.5).

Como é observado, α_i é proporcional à soma de todas as estimativas a posteriori h_i^k dos graus de pertinência de todo o conjunto de treinamento. Portanto, quanto mais cada regra é ativada, maior será o respectivo valor de α_i .

Esta característica é de grande utilidade para a seleção de regras a serem eliminadas. Se existe a necessidade de eliminar alguma regra ou modelo local, um possível candidato será aquele com o menor valor de α_i . O valor mínimo ou limiar para α_i é denotado por α_{min} , de tal forma que, se

$$\alpha_i < \alpha_{min} \quad (5.13)$$

então a i -ésima regra nebulosa será eliminada da estrutura atual do modelo, sendo α_{min} definido pelo usuário via tentativa e erro.

Após a eliminação de uma regra, o **C-FSM** é re-ajustado por algumas iterações EM utilizando as Equações (5.5)–(5.8) até atingir a convergência. Ou seja, até a função de verossimilhança atingir um máximo local ou um máximo número de iterações previamente definido.

A utilização da função de verossimilhança como função objetivo, em conjunto com o algoritmo EM é de interesse neste trabalho, já que este não só trata do erro quadrático

médio, mas também da variância do erro, aumentando assim, as possibilidades de obter melhores resultados.

Ambos os operadores, de adição e de eliminação de regras nebulosas serão também aplicados na estratégia de aprendizado *online* a ser descrito no próximo capítulo.

A primeira etapa de inicialização descrita para esta abordagem de aprendizado *offline* é necessária, já que, o algoritmo EM é sensível à inicialização. Como o algoritmo de aprendizado construtivo *offline* é baseado nas Equações (5.5)-(5.8), as quais são derivadas do algoritmo de otimização EM, uma boa inicialização aumentará as possibilidades de obter modelos eficientes no final do aprendizado construtivo. Assim, o algoritmo de aprendizado construtivo pode ser resumido de acordo com os passo seguinte:

Algoritmo 5 APRENDIZADO CONSTRUTIVO *OFFLINE*

Seja N o número de dados entrada-saída que compõem o conjunto de treinamento.

Definir α_{min} .

Passo 1. Inicialização do modelo **C-FSM**:

- **Passo 1.1** Definir uma estrutura inicial para o modelo, composto por $M = M^0$ regras nebulosas.
- **Passo 1.2** Ajustar os parâmetros utilizando o algoritmo EM tradicional utilizando o conjunto de treinamento previamente definido.

Passo 2. Aprendizado:

- **Passo 2.1** Adição: Verificar a necessidade de uma nova regra utilizando o critério definido pela Equação (5.9). Caso seja necessário, gerar nova regra e re-ajustar os parâmetros do modelo utilizando o algoritmo EM tradicional.
 - **Passo 2.2** Punição: Verificar a relevância de cada uma das regras. Eliminar da estrutura do modelo as regras que satisfaçam o critério de poda definido pela Equação (5.13). Re-ajustar os parâmetros do modelo utilizando o algoritmo EM tradicional.
 - **Passo 2.3** Verificar o critério de parada (ou por número máximo de iterações ou por erro mínimo atingido). Se o critério de parada for satisfeito, para o processo. Caso contrário, voltar para o passo 2.1.
-

5.3 Um exemplo demonstrativo

Como exemplo ilustrativo do processo de ajuste de um modelo de série temporal com a proposta deste capítulo, considerou-se a série de fornalha de gás Box-Jenkins, amplamente utilizada na literatura (Angelov e Filev, 2005).

A série de Box-Jenkins (Box et al., 1994), é composta por um total de 296 dados do processo, sendo os componentes de entrada a taxa de fluxo de gás de alimentação da fornalha, a qual originará como saída uma determinada concentração de dióxido de carbono no gás de escape. Assim, o objetivo do modelo de série temporal é a previsão da concentração de dióxido de carbono (CO_2), com base nos valores passados da mesma e na entrada u^k do modelo. O modelo utilizado adota a seguinte forma:

$$y^k = f(x_1, x_2) \quad (5.14)$$

com $x_1 = y^{k-1}$ e $x_2 = u^{k-4}$, dando assim, um total de 290 dados entrada-saída do processo a modelar, os quais são ilustrados na Figura 6. As entradas neste caso, foram as mesmas utilizadas por outras abordagens na literatura já que posteriormente, os resultados obtidos pelo modelo construtivo **C-FSM** serão comparados aos fornecidos por essas abordagens.

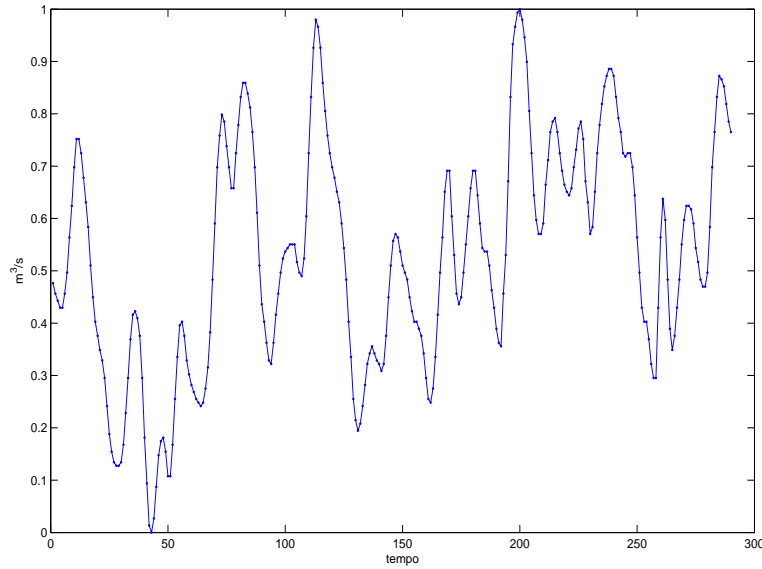


Figura 6: Série de fornalha de gás Box-Jenkins.

Um modelo construtivo é ajustado, sendo o limiar $\alpha_{min} = 0,1$, e critério de parada como

um número máximo de iterações igual a 40. Durante o aprendizado, o máximo número de regras gerado é limitado pelo número de dados disponíveis para o aprendizado. Assim, se um modelo possui p entradas e N dados de treinamento, então, o número máximo de regras nebulosas que pode ser gerado é definido pela seguinte equação:

$$M_{max} = \frac{N - 1}{3p + 1} \quad (5.15)$$

onde M_{max} é o número máximo de regras a ser gerado. Neste caso, o conjunto de treinamento é composto pelos primeiros 200 pares entrada-saída, sendo os últimos 90 pares utilizados na fase de teste do modelo obtido. Assim, o número máximo de regras para a série de fornalha de gás Box-Jenkins é igual a 28.

A Figura 7 ilustra a variação dos centros e parâmetros de dispersão para algumas iterações do processo de aprendizado. A Figura 7-(a) ilustra os centros obtidos na etapa de inicialização, assim como as curvas de nível considerando as matrizes de covariância obtidas para cada uma das regras nebulosas.

A Figura 7-(b) apresenta os centros na primeira iteração da etapa de aprendizado, no qual uma regra nova é gerada e acrescentada na estrutura do modelo de previsão. Uma quarta regra é gerada na segunda iteração (Figura 7-(c)). Com o passar das iterações, a medida que os centros vão se re-allocando, uma das quatro regras nebulosas mantidas até esse instante é eliminada (Figura 7-(d)) devido ao seu valor de α_i que atinge uma magnitude inferior ao limiar α_{min} previamente definido.

Nas próximas iterações a estrutura se mantém com um número constante de regras nebulosas, como ilustra as Figuras 7-(e) e 7-(f), ou seja $M = 3$, variando apenas os parâmetros dos antecedentes e dos conseqüentes. Assim, a estrutura final obtida via o aprendizado construtivo para o problema em estudo, é composto por um total de três regras nebulosas do tipo Takagi-Sugeno definidas como:

$$\begin{aligned} R_1^k : & \text{ Se } < \mathbf{x}^k \in \mathcal{A}_1, g_1^k = f(\alpha_1, \mathbf{c}_1, \mathbf{V}_1) > \text{ Então } < y_i^k = \theta_1 \phi^{\mathbf{k}^T} > \\ R_2^k : & \text{ Se } < \mathbf{x}^k \in \mathcal{A}_2, g_2^k = f(\alpha_2, \mathbf{c}_2, \mathbf{V}_2) > \text{ Então } < y_i^k = \theta_2 \phi^{\mathbf{k}^T} > \\ R_3^k : & \text{ Se } < \mathbf{x}^k \in \mathcal{A}_3, g_3^k = f(\alpha_3, \mathbf{c}_3, \mathbf{V}_3) > \text{ Então } < y_i^k = \theta_3 \phi^{\mathbf{k}^T} > \end{aligned}$$

onde os centros \mathbf{c}_i , a diagonal principal das matrizes de covariância \mathbf{V}_i , os coeficientes dos

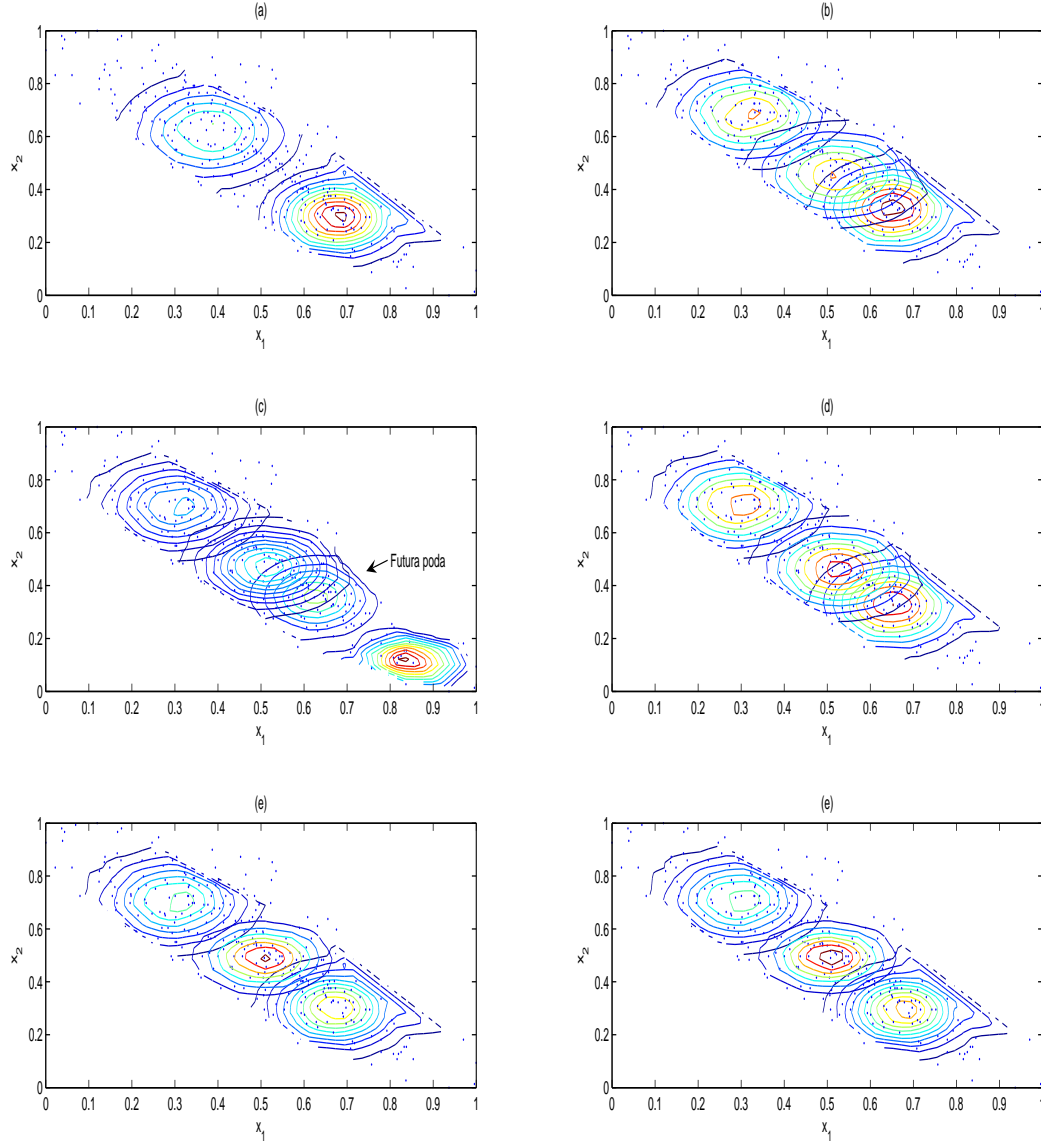


Figura 7: Localização dos centros e curvas de nível associadas para diferentes estágios do processo de aprendizado construtivo *offline*, para o modelo da fornalha de gas Box-Jenkins.

conseqüentes e os coeficientes α_i , para $i = 1, 2, 3$ são definidos como segue:

$$\text{diag}(\mathbf{V}_1) = [0, 020 \ 0, 020]; \quad \text{diag}(\mathbf{V}_2) = [0, 014 \ 0, 015]; \quad \text{diag}(\mathbf{V}_3) = [0, 012 \ 0, 010]$$

$$\mathbf{c}_1 = [0, 293 \ 0, 710]; \quad \mathbf{c}_2 = [0, 678 \ 0, 293]; \quad \mathbf{c}_3 = [0, 504 \ 0, 493]$$

$$\theta_1 = [0, 435 \ -0, 477 \ 0, 592]; \quad \theta_2 = [0, 431 \ -0, 453 \ 0, 571]; \quad \theta_3 = [0, 561 \ -0, 646 \ 0, 538]$$

$$\alpha_1 = 0,304; \quad \alpha_2 = 0,427; \quad \alpha_3 = 0,269$$

Desta forma, cada regra R_i^k codificada na estrutura do modelo construtivo pode ser interpretada da seguinte maneira:

“Se o dado de entrada \mathbf{x}^k pertence a região \mathcal{A}_i , com grau de pertinência g_i^k , então a saída do modelo é igual a $y_i^k = \theta_i \times \phi^{\mathbf{k}^T}$ ”

para $i = 1, 2, 3$.

A Figura 8 ilustra o grau de ativação de cada uma das três regras nebulosas que compõem a estrutura final do modelo de série temporal, para cada um dos dados do conjunto de treinamento. Como pode ser observado, as três regras se complementam, já que a soma dos graus de pertinência para cada dado de entrada é sempre igual a unidade. Assim, enquanto em alguns instantes uma regra é mais importante, em outros ela é menos relevante devido à participação das demais.

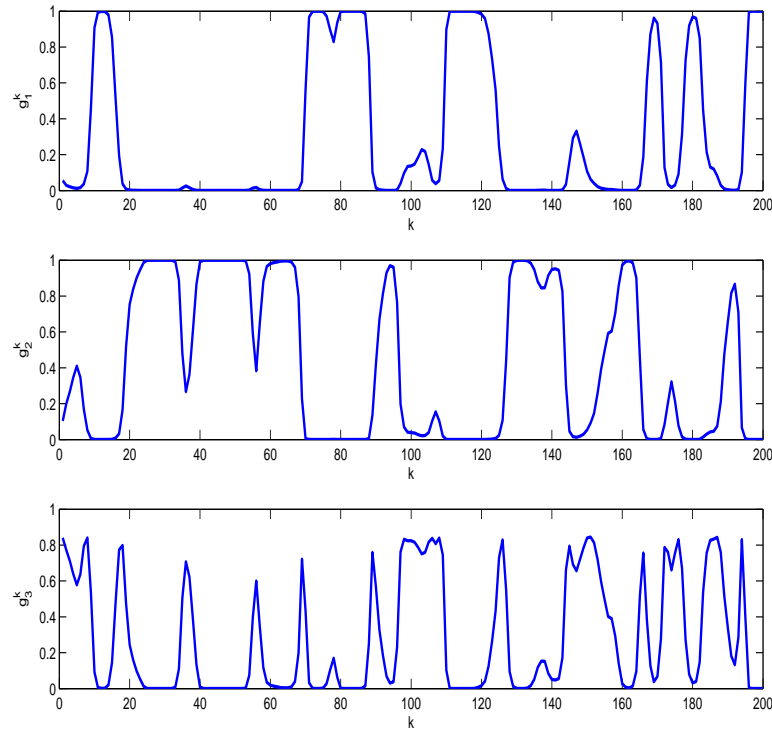


Figura 8: Ativação das regras nebulosas sob o conjunto de treinamento.

5.4 Resumo

Este capítulo apresentou um modelo baseado em regras nebulosas, com aprendizado construtivo *offline*. O modelo construtivo tem como vantagem a geração automática de regras nebulosas para a construção do modelo de série temporal. O algoritmo de aprendizado que foi proposto é baseado no algoritmo da maximização da verossimilhança, assim como nos operadores de geração de novas regras nebulosas e de eliminação ou poda de regras.

A proposta foi aplicada a um problema da literatura bastante utilizado como métrica de comparação de desempenho de novas abordagens. A série representa a concentração de dióxido de carbono da fornalha de gás Box-Jenkins. Esta aplicação teve como objetivo ilustrar a variação da estrutura e dos parâmetros do modelo durante o processo de aprendizado, assim como a ativação das funções de pertinência geradas para o conjunto de treinamento.

6 *Modelo baseado em regras nebulosas com aprendizado online*

6.1 Introdução

O aprendizado adaptativo *online* é uma abordagem com vantagens que vão desde a seleção automática de estrutura de modelo e os seus respectivos parâmetros, até adaptação a mudanças no ambiente (Er e Wu, 2002).

Além disso, algumas técnicas de aprendizado *online* não precisam re-ajustar o modelo globalmente a cada iteração, ou seja, não é necessário o re-ajuste do modelo híbrido ao longo do tempo, utilizando o histórico completo da série a cada vez que um novo dado é disponibilizado.

Evidentemente, é mais fácil re-ajustar um modelo já existente considerando apenas o novo dado de entrada-saída disponível, quando comparado ao re-ajuste do modelo com todo o histórico. No caso da operação e planejamento energético, esta característica é importante já que a complexidade do problema já é alta o suficiente, sendo qualquer abordagem para construção de modelos de séries temporais com menor complexidade computacional é de grande interesse.

Este capítulo propõe uma alternativa de aprendizado *online*, baseado no algoritmo EM e aplicado a modelos baseados em regras nebulosas. Para tal, as Equações (5.5)-(5.8) precisam ser redefinidas, para obter assim versões recursivas das mesmas, de tal forma que a atualização *online* da estrutura e dos parâmetros do modelo seja factível. Para realizar a adaptação da estrutura do modelo, o algoritmo de aprendizado é aplicado em conjunto com os operadores de geração e punição ou eliminação de regras nebulosas descritas no Capítulo 5, os quais também foram adaptadas para a sua utilização em uma versão *online*.

Este capítulo finaliza com a aplicação da proposta do modelo *online* para a construção de um modelo baseado em regras nebulosas para a série de fornalha de gás Box-Jenkins (Box et al., 1994).

6.2 Modelo baseado em regras nebulosas com aprendizado *online* ou seqüencial

Existem trabalhos com idéias similares as descritas neste trabalho, onde o algoritmo EM é re-escrito de forma recursiva, como por exemplo, os trabalhos desenvolvidos por Sato (2000), sendo uma proposta inicial detalhada em (Sato e Ishii, 1999). Paralelamente, a proposta descrita em (Ramamurti e Ghosh, 1996) destaca-se por um algoritmo EM recursivo simplificado, quando comparado à proposta de Sato e Ishii (1999). Destacam-se também as propostas nas quais o algoritmo EM é utilizado para o ajuste de diferentes modelos de redes neurais, como por exemplo o trabalho apresentado por Lázaro et al. (2003).

No entanto, a proposta deste trabalho possui diferenças relevantes. O trabalho descrito por Ramamurti e Ghosh (1996) propõe um algoritmo EM recursivo, o qual é aplicado como um algoritmo de aprendizado para modelos de mistura de especialistas. Neste caso, o algoritmo é aplicado apenas após o ajuste *offline* do modelo, considerando o total de dados disponíveis para o ajuste do modelo. Nesta tese, a construção de modelos baseados em regras nebulosas é realizada por meio de duas etapas bem definidas: uma etapa de inicialização *offline* que utiliza uma parcela pequena de dados, e uma etapa de atualização recursiva ou *online* que continua com o ajuste da estrutura e dos parâmetros do modelo.

Além disso, o enfoque utilizado é diferente aos outros dois trabalhos encontrados na literatura, uma vez que as equações definidas para o ajuste de mistura de especialistas na sua versão *offline* são interpretadas sob o ponto de vista de base de regras nebulosas, sendo as equações adaptadas para a construção de sistemas nebulosos dinâmicos. Assim, é necessário definir operadores de geração e punição de regras nebulosas que possam ser utilizados de forma recursiva, de acordo com o algoritmo de aprendizado a ser aplicado, sendo estes operadores diferentes aos propostos por Ramamurti e Ghosh (1996) e Sato (2000), respectivamente.

O modelo *online* proposto (**A-FSM**), similar ao modelo construtivo *offline* do capítulo anterior, é construído por meio de um processo de modelagem da série. Este processo, como já foi mencionado, consta de uma etapa de inicialização do modelo e de uma segunda etapa de ajuste dos parâmetros e adaptação. A etapa de inicialização do modelo é necessária para gerar um ponto de partida adequado para o aprendizado recursivo.

A etapa de aprendizado ou adaptação é baseada em um algoritmo EM seqüencial, permitindo a adaptação simultânea, tanto da estrutura do modelo, como dos seus parâmetros durante o aprendizado *online*. Ou seja, o modelo é re-ajustado de forma permanente, a medida que novos dados são disponibilizados. Desta maneira, o modelo obtém uma maior flexibilidade e capacidade de adaptação às mudanças na dinâmica do fenômeno em estudo.

6.2.1 Primeira etapa: inicialização

A etapa de inicialização consiste na geração de um ótimo local inicial e factível para o modelo de série temporal. A importância da inicialização para os modelos propostos neste trabalho é que o algoritmo da maximização da verossimilhança é sensível às condições iniciais para o ajuste do modelo. Por este motivo, uma das principais considerações adotadas na construção dos mesmos, é a geração de ótimos locais iniciais factíveis. Para tal, no caso do modelo construtivo, o número máximo de regras a ser gerado foi restrito à condição definida pela Equação (5.15).

No caso do modelo com aprendizado *online*, a condição definida pela Equação (5.15) é verificada previamente à geração de uma nova regra a medida que os dados são apresentados, já que, cada iteração implica no conhecimento de um novo dado, o qual será utilizado para o re-ajuste seqüencial do modelo.

Da mesma forma, esta condição deve ser satisfeita na etapa de inicialização do modelo. Neste trabalho, a inicialização é realizada aplicando dois algoritmos diferentes: o algoritmo construtivo proposto no capítulo anterior (**C-FSM**), e um algoritmo de agrupamento de dados. A principal característica de ambos os algoritmos utilizados nesta etapa de inicialização, é que ambos definem de forma não supervisionada o número de regras nebulosas que compõem a estrutura inicial do modelo. Além disso, estes algoritmos são de natureza *offline*.

O algoritmo de agrupamento utilizado é conhecido como algoritmo SC ou *Subtractive Clustering*, o qual foi originalmente proposto em (Chiu, 1994). Este algoritmo de agrupamento é baseado no algoritmo Fuzzy C-Means (Bezdek, 1981) e no algoritmo de clusterização da Montanha ou *Mountain Clustering* (Yager e Filev, 1994).

Assim, a inicialização do modelo é atingida da seguinte maneira. O conjunto total de dados disponíveis é dividido em dois sub-conjuntos. O primeiro sub-conjunto é composto por uma pequena parte inicial do histórico da série, sendo utilizado na etapa de inicialização *offline*. O total de dados que compõem o conjunto de dados de inicialização é representado por N^0 . O segundo sub-conjunto composto pelo restante do histórico será utilizado na parte de

ajuste *online* do modelo, ou seja, o ajuste seqüencial será aplicado para $k = N^0 + 1, N^0 + 2, \dots$

No caso da inicialização ser obtida utilizando o **C-FSM**, o número de regras nebulosas, assim como os valores iniciais dos parâmetros são fornecidos pelo próprio algoritmo construtivo, utilizando os primeiros N^0 dados da série. Este algoritmo foi detalhado no Capítulo 5.

Por outro lado, o algoritmo de agrupamento SC fornece o número inicial de regras e os centros associados. Porém, é necessário obter um modelo inicial, com valores iniciais para todos os parâmetros do modelo, incluindo matrizes de covariância e os coeficientes dos conseqüentes. Por este motivo, após a obtenção dos centros e do número inicial de regras nebulosas é necessário fazer um ajuste do modelo, que considere todos os parâmetros do mesmo. O modelo com aprendizado online e etapa de inicialização utilizando o algoritmo SC é denominado **mA-FSM**. O algoritmo SC é descrito brevemente a seguir.

6.2.1.1 O algoritmo de agrupamento subtrativo - SC

O algoritmo SC proposto por Chiu (1994), vem sendo um algoritmo bastante empregado para a geração de modelos baseados em regras, devido a sua simplicidade e capacidade para determinar de forma automática o número de centros necessários para representar de forma adequada o conjunto de dados, o qual é uma grande vantagem quando comparado a outros algoritmos de agrupamento, como por exemplo, o algoritmo tradicional *Fuzzy C-Means* (FCM) (Bezdek, 1981).

A idéia base do algoritmo SC é a seguinte. O algoritmo define um índice de potencial P_k para cada dado entrada-saída, com $k = 1, \dots, N^0$. O potencial de cada dado depende diretamente da distância deste para todos os outros dados do conjunto apresentado. Assim, quanto menor for a distância do dado aos outros, seu potencial será maior. Desta forma, o potencial pode ser definido como:

$$P_k = \sum_{j=1}^N \exp\{-\alpha \|\psi_k - \psi_j\|^2\} \quad (6.1)$$

com $\psi_k = [\mathbf{x}^k \ y^k]$, $\alpha = 4/r_a^2$, onde \mathbf{x}^k é o vetor de entrada ao modelo composto por p componentes; y^k é a saída unidimensional e r_a é um termo constante e positivo.

Deve-se observar que, todo dado que se pertença a uma região com uma densidade alta de dados, terá um potencial maior, tendo assim, uma possibilidade mais alta de se tornar um novo centro. Além disso, o parâmetro r_a definirá o raio de influência na vizinhança de cada

um dos centros, ou seja, a região na qual os centros exercerão a sua influência.

Seja ψ_1 o dado entrada-saída com o maior potencial P_1^* . Assim, o potencial para todos os outros dados serão atualizados como:

$$P_k = P_k \times (1 - \exp\{-\beta\|\psi_k - \psi_1\|^2\}) \quad (6.2)$$

com $\beta = 4/r_b^2$ e $r_b = 1,5r_a$. Desta maneira, quanto mais próximo esteja um dado entrada-saída de um centro já definido, menor será o seu potencial, sendo menor a sua possibilidade de se tornar um novo centro. O Algoritmo 6 resume os passos que devem ser seguidos para a definição dos centros.

Algoritmo 6 Algoritmo de agrupamento subtrativo - SC

Sejam N^0 dados entrada-saída construídos com uma parte do histórico da série disponível para a etapa de inicialização do modelo, $N^0 \ll N$, onde N é o total de dados entrada-saída construídos a partir do histórico completo da série. Definir o valor de r_a . Calcular o potencial de todos os N^0 dados entrada-saída utilizando a Equação (6.1) e selecionar o primeiro centro como sendo igual ao padrão com maior potencial (P_1^*).

Passo 1. Atualizar os potenciais P_k de todos os dados utilizando a Equação (6.2).

Passo 2. Verificar qual é o dado com potencial mais alto;

Passo 3. Se $P_k^* > \bar{\varepsilon}P_1^*$, com $\bar{\varepsilon} = 0,5$, então ψ_k^* é definido como um novo centro, ir ao Passo 5. Senão, ir ao Passo 4.

Passo 4. Se $P_k^* < \underline{\varepsilon}P_1^*$, com $\underline{\varepsilon} = 0,15$, então o possível centro é rejeitado e o processo termina. Senão, ir ao Passo 6.

Passo 5. Atualizar os potenciais utilizando a Equação (6.2) e voltar ao Passo 2.

Passo 6. Seja d_{min} a distância mínima entre ψ_k^* e todos os outros centros já selecionados. Se

$$\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$$

então ψ_k^* é definido como um novo centro e retorne ao Passo 5. Senão, ψ_k^* é rejeitado e $P_k^* = 0$. Selecionar o próximo dado com maior potencial e defini-lo como um possível novo centro. Verificar novamente a condição do Passo 6.

Seja M^0 o número de regras iniciais definido pelo algoritmo SC. Para este número inicial de regras M^0 , os parâmetros do modelo são inicializados da seguinte forma:

- $\mathbf{c}_i^0 = \psi_i^0|_{1\dots p}$, onde $\psi_i^0|_{1\dots p}$ é composto pelos primeiros p componentes do i -ésimo centro definido pelo algoritmo SC;
- $\sigma_i^0 = 1, 0$;
- $\theta_i^0 = [\psi_i^0|_{p+1} \ 0 \ \dots \ 0]_{1 \times p+1}$, onde $\psi_i^0|_{p+1}$ é o $p + 1$ -th componente do i -ésimo centro encontrado pelo algoritmo SC;
- $\mathbf{V}_i^0 = 10^{-4}\mathbf{I}$, onde \mathbf{I} é uma matriz identidade de ordem p ;
- $\alpha_i^0 = 1/M^0$.

Após a inicialização dos parâmetros procede-se o ajuste do modelo, aplicando uma seqüência de passos EM para a otimização dos parâmetros, considerando não apenas os centros e sim os parâmetros de dispersão e os coeficientes que definem os conseqüentes das M^0 regras nebulosas (ver Algoritmo 4, Capítulo 5). Desta forma, obtém-se um ótimo local factível para dar início ao ajuste online.

6.2.2 Segunda etapa: adaptação

Durante o aprendizado *online*, a adaptação da estrutura e parâmetros será executada considerando uma janela no tempo, representada por T . Isto é, os últimos dados T influenciarão fortemente os parâmetros do modelo e a sua estrutura, mas o histórico todo contribuirá como um componente no cálculo dos parâmetros para a próxima iteração.

O aprendizado adaptativo é realizado a cada instante de tempo k , $k = 1, 2, \dots$. Neste tipo de aprendizado as informações mais recentes têm uma maior influência no ajuste dos parâmetros e na estrutura do modelo. Observando a Eq. (5.5), e considerando uma janela no tempo $T = N$, esta equação pode ser re-escrita como segue:

$$\begin{aligned}
 \alpha_i^N &= \frac{1}{N} \sum_{k=1}^N h_i^k = \frac{1}{N} \left(\sum_{k=1}^{N-1} h_i^k + h_i^N \right) \\
 &= \frac{1}{N} [(N-1)\alpha_i^{N-1} + h_i^N] \\
 &= \alpha_i^{N-1} + \frac{1}{N} [h_i^N - \alpha_i^{N-1}]
 \end{aligned} \tag{6.3}$$

onde h_i^N é a estimação a posteriori de g_i^k e α_i^{N-1} é o valor estimado para α_i considerando apenas os $N - 1$ dados iniciais. Como pode ser observado, a Eq. (6.3) é uma estimação recursiva

de (5.5). Seguindo este mesmo procedimento para um número genérico de iterações k e com uma janela no tempo T , as Eqs. (5.5)-(5.4) podem ser re-escritas recursivamente, obtendo as seguintes aproximações:

$$\alpha_i^{k+1} = \alpha_i^k + \frac{1}{T}[h_i^k - \alpha_i^k] \quad (6.4)$$

$$\mathbf{c}_i^{k+1} = \mathbf{c}_i^k + \frac{1}{\gamma_i^{k+1}}[\mathbf{x}^k - \mathbf{c}_i^k] \quad (6.5)$$

$$\mathbf{V}_i^{k+1} = \mathbf{V}_i^k + \frac{1}{\gamma_i^{k+1}}[(\mathbf{x}^k - \mathbf{c}_i^k)(\mathbf{x}^k - \mathbf{c}_i^k)' - \mathbf{V}_i^k] \quad (6.6)$$

$$(\sigma_i^2)^{k+1} = (\sigma_i^2)^k + \frac{1}{\gamma_i^{k+1}}[(y^k - y_i^k)^2 - (\sigma_i^2)^k] \quad (6.7)$$

nas quais:

$$\frac{1}{\gamma_i^{k+1}} = \frac{h_i^{k+1}}{\sum_{t=1}^{k+1} h_i^t} \quad (6.8)$$

Uma aproximação de $\sum_{t=1}^{k+1} h_i^t$ pode ser construída considerando a mesma janela T e a equação recursiva inspirada no aprendizado adaptativo descrito em (Wang, 1994). Seja:

$$S_i^{k+1} = \sum_{t=1}^{k+1} h_i^t \quad (6.9)$$

Logo, S_i^{k+1} pode ser estimado como:

$$S_i^{k+1} \approx S_i(\mathbf{x}^{k+1}) + \frac{T-1}{T} S_i^k \quad (6.10)$$

o qual pode ser re-escrito da seguinte maneira:

$$S_i^{k+1} \approx S_i^k + \left[S_i(\mathbf{x}^{k+1}) - \frac{S_i^k}{T} \right] \quad (6.11)$$

Na Equação (6.11) nota-se que o termo S_i^k/T pode ser interpretado como uma estimativa do valor médio de α_i dentro da janela de tempo T . Assim, quanto maior for o valor de S_i^k/T , maior será a relevância da i -ésima regra nebulosa para a próxima iteração. Se $S_i(\mathbf{x}^{k+1})$ é baixo em T , então S_i^{k+1} diminuirá e a i -ésima regra terá maior chance de ser eliminada.

Para estimar θ_i , é utilizado o algoritmo de Mínimos Quadrados Recursivos Ponderado ou *Weighted Recursive Least Square Algorithm* - wRLS, o qual considera um fator de esquecimento no tempo, chamado de f_{forget} , como descrito em (Haykin, 2001a). As equações do wRLS, adaptado ao problema em estudo são definidas por:

$$\theta_i^{k+1} = \theta_i^k + \mathbf{C}_i^{k+1} \phi^{\mathbf{k}} \times h_i^k(y^k - y_i^k) \quad (6.12)$$

onde:

$$\mathbf{C}_i^{k+1} = \frac{\mathbf{C}_i^k}{f_{forget}^k + h_i^k(\phi^k)^T \mathbf{C}_i^k \phi^k} \quad (6.13)$$

é a matriz de covariância associada a cada θ_i durante o aprendizado adaptativo online e $f_{forget} \in (0, 1]$ é o fator de esquecimento.

As condições iniciais para θ_i^0 , $i = 1, \dots, M$ são dadas pelos valores obtidos durante a inicialização do modelo, enquanto $\mathbf{C}_i^0 = \mu \mathbf{I}$, com $\mu = 10^4$ e \mathbf{I} é a matriz identidade de tamanho $(p + 1) \times (p + 1)$.

Após a inicialização do modelo a adaptação *online* é aplicada, sendo os parâmetros do modelo modificados de forma permanente e seqüencial. Além disso, a estrutura a cada iteração é avaliada, verificando as condições que requerem a geração de novas regras na estrutura, assim como a eliminação ou poda de regras pouco relevantes para as próximas iterações. Com o objetivo de atualizar a estrutura do modelo, utiliza-se operadores de geração e eliminação de regras nebulosas, de forma similar aos aplicados no algoritmo de aprendizado construtivo *offline*, mas utilizando as Equações (6.4)-(6.13) para tal fim.

A cada novo dado, verificam-se as condições para gerar novas regras, assim como para eliminar regras nebulosas. Embora os critérios de geração e de poda de regras utilizados para o ajuste dinâmico do modelo sejam os mesmos que no aprendizado *offline*, algumas considerações devem ser feitas. Ambos os operadores adaptados para a sua utilização no ajuste recursivo serão detalhados a seguir.

6.2.2.1 Geração de novas regras

Neste caso, a geração de uma nova regra nebulosa e a sua inclusão na estrutura do modelo durante o aprendizado online será efetuada quando o dado atual de entrada \mathbf{x}^k apresentado ao modelo não pertencer a nenhuma região de ativação das regras já existentes. Ou seja a probabilidade do dado de entrada no instante k ativar alguma das regras nebulosas é inferior a um limiar previamente definido.

Da mesma forma que no caso do modelo construtivo, a probabilidade de \mathbf{x}^k ativar alguma das regras nebulosas (ver Equação (4.4)) deve ser superior a 0,1357, de acordo com a tabela de distribuição normal e considerando um nível de confiança de 72,86%. A condição definida pela Equação (5.9) deve também ser satisfeita, sendo a condição verificada apenas para o

dado atual já que não existe um conjunto de dados de treinamento, como no caso do modelo C-FSM.

Caso esta condição não seja satisfeita pelo k -ésimo dado de entrada apresentado uma nova regra é acrescentada ao sistema, sendo os seus parâmetros inicializados da seguinte maneira:

- $\mathbf{c}_{M+1}^{k+1} = \mathbf{x}^k$
- $\sigma_{M+1}^{k+1} = 1, 0;$
- $\theta_{M+1}^{k+1} = [y^k \ 0 \ \dots \ 0]_{1 \times p+1}$
- $\mathbf{V}_{M+1}^{k+1} = 10^{-4} \mathbf{I}$, onde \mathbf{I} é uma matriz identidade de tamanho $p \times p$;
- $\alpha_{M+1}^{k+1} = 10^{-5}$. Embora este valor seja pequeno para interferir na dinâmica da estrutura atual, todos os parâmetros α_i , $i = 1, \dots, (M+1)$ são re-normalizados de tal forma que a soma de todos estes coeficientes seja sempre igual à unidade.

6.2.2.2 Eliminação ou poda

O operador de punição ou eliminação de regras nebulosas para o aprendizado *online* requer mais de uma condição a ser verificada antes da sua aplicação.

Após uma nova regra ser criada, o seu respectivo α_i terá um valor pequeno e igual a 10^{-5} . Se o operador de poda é aplicado imediatamente, esta regra nebulosa recém gerada será eliminada sem a devida verificação da sua relevância na estrutura do modelo.

Esta situação é resolvida com a criação de um novo índice, chamado de índice de permanência τ . Toda vez que uma nova regra é gerada, o seu respectivo índice de permanência τ_i também será criado. A medida que esta regra é ativada ao longo do tempo, este índice é incrementado:

$$\tau_i^{k+1} = \tau_i^k + 1 \quad (6.14)$$

Logo, uma regra nebulosa na estrutura poderá ser eliminada somente se o seu α_i possui um valor muito pequeno:

$$\alpha_i < \alpha_{min}$$

e se

$$\tau_i^k > \epsilon T \quad (6.15)$$

sendo $\epsilon > 0$ e T a janela de tempo utilizada durante o treinamento *online*.

Esta condição garante que nenhuma regra seja eliminada imediatamente após à sua criação, permitindo-lhe um tempo mínimo para a sua adaptação igual a ϵT , eliminando assim, possíveis oscilações desnecessárias e pouco eficientes na estrutura do modelo.

Após a verificação dos critérios de geração de novas regras e de poda, os parâmetros dos modelos são re-ajustados utilizando as equações recursivas e fornecendo a saída estimada para a próxima iteração, a qual é a estimacão conhecida como previsão um passo à frente na teoria de análise de séries temporais (Aguirre, 2004). O algoritmo de aprendizado seqüencial proposto é resumido pelo pseudo-código a seguir:

Algoritmo 7 APRENDIZADO ADAPTATIVO *ONLINE*

Seja $N^0 \ll N$ o número de dados entrada-saída que compõem o conjunto de inicialização, onde N é o total de dados disponíveis.

Definir α_{min} , T , f_{forget} e ϵ .

Passo 1. Inicialização do modelo **A-FSM**/**mA-FSM**:

- **Passo 1.1** Definir uma estrutura inicial para o modelo utilizando o algoritmo de aprendizado construtivo *offline* para o **A-FSM** e o algoritmo SC para o **mA-FSM**.

Passo 2. Aprendizado *online* ou seqüencial:

Para $k = N^0 + 1, \dots$:

- **Passo 2.1** Adição: Verificar a necessidade de uma nova regra utilizando o critério definido pela Equação (5.9) a cada k . Caso seja necessário, gerar nova regra Atualizar $M^{k+1} = M^k = M + 1$.
 - **Passo 2.2** Punição: Verificar a relevância de cada uma das regras. Eliminar da estrutura do modelo as regras que satisfaçam o critério de poda. Re-ajustar os parâmetros do modelo. Atualizar $M^{k+1} = M^k - 1$.
 - **Passo 2.3** Atualizar os parâmetros do modelo utilizando as equações recursivas (6.4)-(6.13). Fazer $k = k + 1$ e calcular a estimativa da saída para o próximo passo (\hat{y}^{k+1}).
-

6.3 Exemplo demonstrativo

Nesta seção, o modelo baseado em regras nebulosas com aprendizado *online* é utilizado para construir um modelo de série temporal variante no tempo, para o mesmo problema

descrito na Seção 5.3, correspondente a série de fornalha de gás Box-Jenkins (Box et al., 1994). Embora o aprendizado seja realizado de forma seqüencial, a estrutura geral do modelo continua sendo representada pela Equação (5.14), com a diferença de $f(\cdot)$ ser variável no tempo.

Neste caso, o modelo foi inicializado utilizando o algoritmo SC. Assim, a etapa de inicialização é realizada, considerando $N^0 = 20$. O número inicial de regras nebulosas encontradas pelo algoritmo SC foi igual a $M^0 = 2$ com $r_a = 0,50$ e $r_b = 1,50$. Assim, o aprendizado *online* inicia para $k = 21, \dots$, com uma janela de tempo T igual a 20, $\alpha_{min} = 0,01$, $\epsilon = 0,50$ e fator de esquecimento $f_{forget} = 0,95$.

O número de regras oscilou entre um mínimo de $M = 2$ e um máximo de $M = 3$ no início da segunda etapa. Deve também ser observado que, embora o número de regras nebulosas permaneça constante para $k > 100$, isso não implica que novas regras não foram criadas enquanto outras estão sendo eliminadas da estrutura do modelo na mesma iteração. A Figura 9 ilustra a variação dos centros e parâmetros de dispersão durante o aprendizado seqüencial.

As curvas de nível representam as regiões de ativação no espaço de entrada, de cada uma das regras nebulosas que compõem a estrutura da base de regras. A nuvem de pontos em azul representam os dados de entrada utilizados até a iteração k . A nuvem de pontos em preto representa os últimos T dados de entrada apresentados ao modelo. A Figura 9-(a) ilustra os centros e parâmetros de dispersão obtidos na etapa de inicialização do modelo. Neste caso, a nuvem de pontos em azul representa o conjunto de dados utilizado na etapa de inicialização, ou seja, os dados para $k = 1, \dots, 20$.

A Figura 9-(b) ilustra as condições para os centros e parâmetros de dispersão para $k = 25$, onde observa-se que uma nova regra é acrescentada na estrutura do modelo, com centro localizado no dado de entrada distinto dos dados apresentados até esse instante. Os três centros da Figura 9-(b) possuem as seguintes coordenadas:

$$\mathbf{c}_1^{25} = [0,602 \ 0,419]; \quad \mathbf{c}_2^{25} = [0,324 \ 0,677]; \quad \mathbf{c}_3^{25} = [0,810 \ 0,242]$$

sendo \mathbf{c}_3 o novo centro que representa a terceira regra recém gerada e codificada na estrutura. No instante $k = 34$, uma das regras é eliminada da base, como ilustra a Figura 9-(c). Comparando esta ilustração com a Figura 9-(b), observa-se que a regra associada a \mathbf{c}_3 re-alocou o seu centro, e modificou a sua matriz de covariância, incluindo assim a região ativa associada a \mathbf{c}_1 , não tendo mais a necessidade da regra associada à sub-região com centro em \mathbf{c}_1 , como ilustrado na Figura 9-(c).

A Figura 9-(d), (e) e (f) ilustram as condições dos centros e parâmetros de dispersão para $k = 120$, 170 e $k = 220$, respectivamente.

Destas últimas três ilustrações, nota-se que os centros vão se deslocando para diferentes direções, os quais dependem dos dados apresentados recentemente. Em outras palavras, os últimos T dados (nuvem de pontos pretos), influenciam a localização dos centros no tempo, e portanto, das regiões de ativação das regras nebulosas.

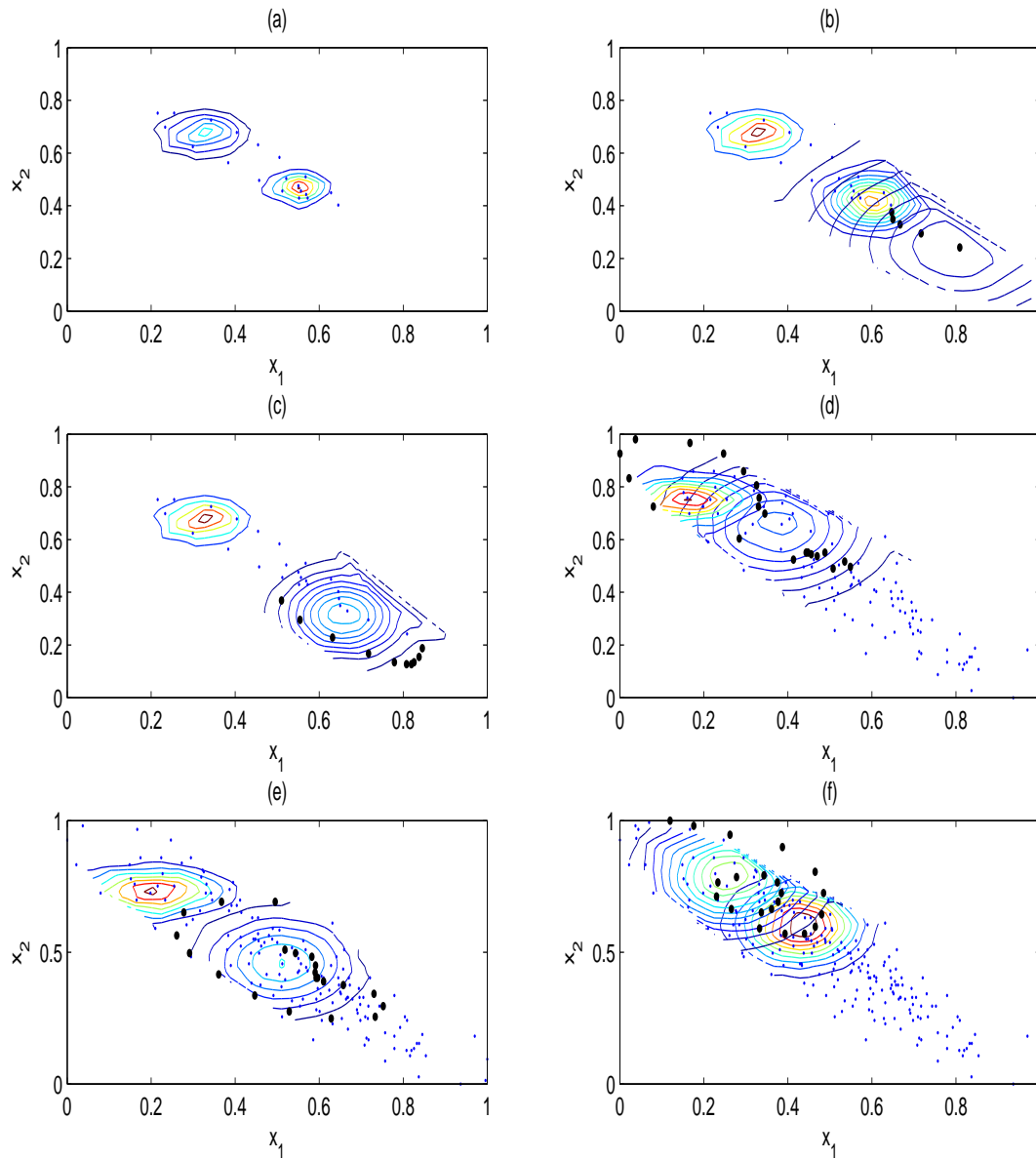


Figura 9: Localização dos centros e curvas de nível associadas para diferentes estágios do processo de aprendizado sequencial, para o modelo da fornalha de gás Box-Jenkins.

6.4 Resumo

Este capítulo apresentou uma proposta de modelo dinâmico baseado em regras nebulosas. O processo de aprendizado seqüencial ou dinâmico é composto de duas etapas. A etapa de inicialização tem por objetivo a geração de uma estrutura inicial factível de base de regras nebulosas, a qual será modificada e os seus parâmetros otimizados durante a segunda etapa de aprendizado.

O algoritmo de aprendizado utilizado é baseado em uma abordagem recursiva, a qual é aplicada em conjunto com operadores de geração e eliminação ou poda de regras nebulosas.

O exemplo apresentado ilustra brevemente a dinâmica do aprendizado, assim como a atualização da estrutura do modelo para a série de fornalha de gás Box-Jenkins.

7 *Modelagem e previsão de séries temporais*

7.1 Introdução

Este capítulo apresenta um conjunto de testes comparativos entre os modelos propostos neste trabalho e outros modelos da literatura. Os modelos comparativos foram escolhidos para cada aplicação, devido a sua relevância na área de pesquisa. Em outros casos, os modelos foram selecionados por serem amplamente utilizados para os problemas em estudo.

7.2 Série de fornalha de gás Box-Jenkins

A série de fornalha de gás Box-Jenkins vem sendo utilizada a partir do Capítulo 5, como exemplo demonstrativo durante o aprendizado construtivo e *online* dos modelos baseados em regras nebulosas. Como foi descrito no Capítulo 5, a série de fornalha de gás Box-Jenkins é composta por 290 dados entrada-saída, construídos a partir de um total de 296 dados do histórico da série, o qual foi obtido mediante testes de laboratório e apresentado em (Box et al., 1994).

A variável de entrada u^k e saída desejada y^k representam, respectivamente, o fluxo de metano e a concentração de CO_2 . A estrutura do modelo de série temporal é construído utilizando as variáveis de entrada u^{k-4} e y^{k-1} , ou seja:

$$\hat{y}^k = f[y(k-1), u(k-4)] \quad (7.1)$$

onde \hat{y}^k é a saída estimada do modelo baseado em regras nebulosas. As entradas mencionadas são as mesmas utilizadas por Angelov e Filev (2005), e serão as mesmas utilizadas neste trabalho de tal forma que a comparação dos resultados seja possível. O modelo proposto

em (Angelov e Filev, 2005) é chamado de **simplLeTS**. Este modelo trata-se também de um modelo baseado em regras nebulosas do tipo Takagi-Sugeno, com aprendizado *online*. Os resultados obtidos pelos modelos **C-FSM**, **A-FSM** e **mA-FSM** são comparados aos obtidos pelos modelos **eTS**, assim como o melhor resultado obtido pelo **simplLeTS**, os quais são detalhados em (Angelov e Filev, 2005).

Os indicadores de desempenho utilizados para esta aplicação são a raiz do erro quadrático médio (**REQM**) e o índice de erro não-dimensional **NDEI** (*non-dimensional error index*). Estas medidas de erro são definidas por:

$$\text{REQM} = \left(\frac{1}{N} \sum_{k=1}^N (y^k - \widehat{y}^k)^2 \right)^{\frac{1}{2}} \quad (7.2)$$

$$\text{NDEI} = \frac{\text{REQM}}{\text{std}(y^k)} \quad (7.3)$$

sendo N o número de dados a prever, \widehat{y}^k a saída do modelo, y^k a saída desejada e $\text{std}(\cdot)$ o desvio padrão.

A etapa de inicialização dos modelos com aprendizado seqüencial (**A-FSM** e **mA-FSM**) é realizada considerando apenas os primeiros 20 dados entrada-saída disponíveis, ou seja, $N^0 = 20$. No caso do modelo construtivo **C-FSM**, a inicialização foi efetuada utilizando os primeiros 200 dados entrada-saída, os quais também serão utilizados na etapa de aprendizado.

A estrutura inicial do modelo construtivo é composta por duas regras nebulosas ajustadas utilizando o algoritmo EM tradicional. A etapa de aprendizado construtivo foi desenvolvida considerando $\alpha_{\min} = 0,1$ e um máximo de iterações igual a 40 como critério de parada.

A estrutura final do **C-FSM** é composta por três regras nebulosas, com parâmetros definidos no Capítulo 5, Seção 5.3.

O número inicial de regras nebulosas obtidas utilizando o algoritmo SC para o modelo com aprendizado seqüencial foi igual a $M^0 = 2$ com $r_a = 0,25$ e $r_b = 1,50$. Assim, o aprendizado seqüencial foi iniciado para $k = 21, \dots$, com $T = 20$, $\alpha_{\min} = 0,01$, $\epsilon = 0,50$ e $f_{\text{forget}} = 0,95$.

A Figura 10 ilustra a variação do número de regras nebulosas durante o aprendizado dos modelos. A Figura 10-(a) ilustra a variação no tamanho da estrutura durante o aprendizado *offline* para o modelo **C-FSM**. A variação dos modelos com aprendizado seqüencial **A-FSM** e **mA-FSM** são apresentados nas Figuras 10-(b) e 10-(c), respectivamente, para $k = 21, \dots$

Deve-se notar que, embora o número inicial de regras nebulosas seja igual para o modelo **A-FSM** e **mA-FSM**, a variação da estrutura durante o aprendizado sequencial é diferente para cada modelo, pois os valores iniciais dos parâmetros são diferentes para cada estrutura inicial. Esta diferença na variação na estrutura ilustra a sensibilidade do algoritmo de aprendizado, para os valores iniciais considerados, o que conseqüentemente reflete a importância da etapa de inicialização do modelo.

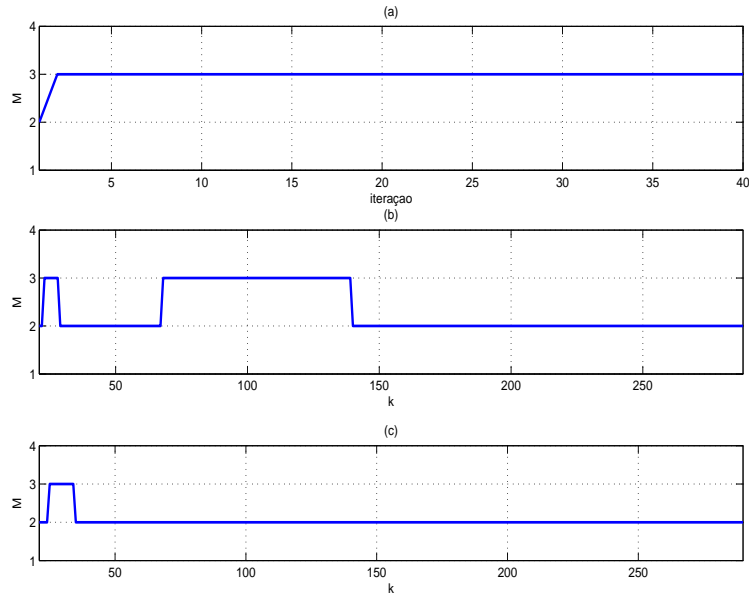


Figura 10: Variação de M durante o aprendizado dos modelos baseados em regras para a série de fornalha de gás Box-Jenkins: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

Os erros de previsão um passo à frente obtidos são apresentados na Tabela 2.

Tabela 2: Erros de previsão um passo à frente para a série de fornalha de gás Box-Jenkins

Modelo	M^0 inicial	M teste	REQM	NDEI
C-FSM	2	3	0,0482	0,3005
A-FSM	2	2	0,0322	0,2006
mA-FSM ($r_a = 0.25$)	2	2	0,0322	0,2006
eTS (Angelov e Filev, 2005)	0	5	0,0490	0,3057
simplLeTS (Angelov e Filev, 2005)	0	3	0,0485	0,3004

Os resultados de simulação mostram que os modelos propostos neste trabalho são adequados para o problema em estudo, obtendo desempenhos similares ou melhores aos obtidos

pelos modelos propostos em (Angelov e Filev, 2005), com estruturas compactas.

Na Tabela 2 nota-se que, tanto o modelo **eTS** como o **simpleTS** podem inicializar as suas estruturas sem nenhuma informação a priori, ou seja, sem nenhuma regra e com somente um dado entrada-saída.

Deve se observar que, para fins de comparação, a série temporal não foi transformada para a sua versão estacionária, dado que os modelos propostos por Angelov e Filev (2005) trabalham com a série original, ou seja, sem retirar os componentes de tendência ou sazonalidade que possam existir, sendo esta normalizada apenas no intervalo $[0, 1]$.

Para esta série, observou-se que o modelo com aprendizado *online* fornece melhores resultados de previsão quando comparado ao modelo construtivo **C-FSM**, já que os modelos **A-FSM** e **mAS-FSM** têm a capacidade de se adaptar a mudanças na dinâmica da série. Os resultados de previsão um passo à frente são ilustrados na Figura 11.

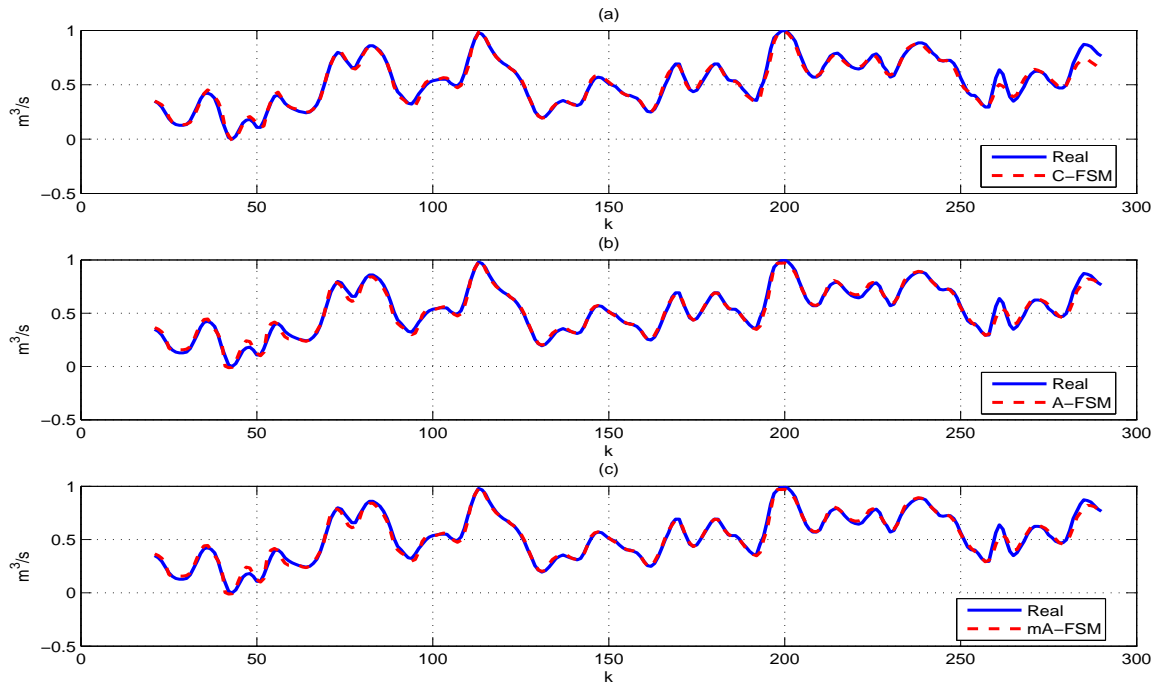


Figura 11: Previsão um passo à frente da série de fornalha de gás Box-Jenkins: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

7.3 Série econômica

Nesta aplicação, o desempenho dos modelos **C-FSM**, **A-FSM** e **mAS-FSM** é verificado usando dados reais de uma série econômica da taxa de desemprego trimestral dos Estados Unidos. Estes dados foram utilizados em (Montgomery et al., 1998) e em (Tsay, 2005) para a construção e avaliação de modelos lineares e não lineares de regressão. Neste caso, os resultados obtidos na previsão de um até cinco passos à frente, são comparados com os resultados obtidos em (Tsay, 2005), onde foi ajustado o modelo linear **ARIMA** e os modelos não-lineares **TAR** e **MSA**.

O histórico da série é composto por dados a partir do primeiro trimestre de 1948 (1948-I), até o quarto trimestre de 1993 (1993-IV). Esta série, sazonalmente ajustada, é calculada como a média ponderada das taxas de desemprego mensais para cada trimestre, sendo analisada de forma detalhada em (Montgomery et al., 1998).

Algumas características dessa série são: a série possui um comportamento cíclico altamente relacionado aos ciclos econômicos dos Estados Unidos. Estes ciclos começam com um crescimento acelerado da taxa de desemprego e finaliza com picos elevados, os quais são alternados com quedas menos acentuadas. Estas características sugerem que a dinâmica da série é de natureza não-linear.

O primeiro passo para a construção do modelo de série temporal é a seleção de entradas. Neste sentido, e para comparar o desempenho dos modelos, as entradas utilizadas neste trabalho são as mesmas utilizadas pelos modelos não lineares **TAR** e **MSA** em (Tsay, 2005). Assim, foi necessário trabalhar com a série estacionária. Por tal motivo, considerou-se a primeira diferença da série $\Delta x^k = x^t - x^{k-1}$, retirando antes o componente sazonal, sendo as entradas iguais a Δx^{k-1} e Δx^{k-2} . Desta forma, os modelos de série temporal ajustados **C-FSM**, **A-FSM**, **mA-FSM**, **MSA** e **TAR** adotaram a seguinte forma:

$$\hat{y}^k = f[\Delta x^{k-1}, \Delta x^{k-2}] \quad (7.4)$$

onde $f[\cdot]$ é uma função linear de Δx^{k-1} e Δx^{k-2} , quando definida pelos modelos auto-regressivos, e uma função não-linear Δx^{k-1} e Δx^{k-2} quando definida pelos modelos baseados em regras nebulosas.

Para verificar o efeito da transformação da série, um segundo modelo construtivo foi ajustado trabalhando diretamente com a série original transformada apenas linearmente para o intervalo $[0,1]$. Neste segundo caso, o algoritmo FNN-PMI foi utilizado para a seleção de

entradas, dando finalmente como entradas ao modelo, os atrasos 1, 3 e 4, a partir de um conjunto inicial de possíveis entradas de tamanho 4, definida inicialmente pelo algoritmo FNN. Ou seja, o modelo construtivo denominado de **C-FSM_t**, adotou a seguinte forma:

$$\hat{y}^k = f_t[y^{k-1}, y^{k-3}, y^{k-4}] \quad (7.5)$$

onde $f_t[\cdot]$ denota o modelo construtivo ajustado para a série original.

Após definir as entradas dos modelos, as estruturas e parâmetros são determinados de acordo com os algoritmos de aprendizado descritos nos Capítulos 5 e 6.

Ambos os modelos construtivos foram ajustados utilizando dados do histórico desde 1948-I até 1968-II, dando um total de 79 dados entrada-saída no conjunto de treinamento ou calibração para o modelo **C-FSM** e 78 pares entrada-saída para o ajuste do modelo **C-FSM_t**. O valor de α_{min} utilizado para ambos os modelos foi definido como $\alpha_{min} = 0,01$, sendo o aprendizado finalizado após o critério de parada ser atingido. Neste caso, o critério de parada é igual a um número máximo de épocas igual a 40.

No caso dos modelos construtivos, observou-se uma variação no número de regras entre 2 e 4, sendo a estrutura final composta por 4 regras nebulosas no caso do modelo **C-FSM** construído para a série estacionária e 3 regras nebulosas no caso do modelo **C-FSM_t** para a série não estacionária.

Por outro lado, os modelos com aprendizado seqüencial, foram inicializados considerando apenas os primeiros quatro anos do histórico da série, ou seja, a etapa de inicialização dos modelos **A-FSM** e **mA-FSM** utilizou apenas 20 dados entrada-saída para a obtenção de uma estrutura inicial. O algoritmo SC utilizado na fase de inicialização do **mA-FSM** utilizou $r_a = 0,75$ e $r_b = 1,75r_a$. Após a inicialização, procedeu-se o ajuste seqüencial sob os próximos 59 valores do histórico da série. Em ambos os casos, considerou-se $\alpha_{min} = 0,01$, $f_{forget} = 0,95$, $T = 20$ e $\epsilon = 0,50$. Ambos os modelos não sofreram consideráveis variações no tamanho da estrutura. O modelo **A-FSM** manteve o número de regras na sua estrutura constante e igual a 2, enquanto o modelo **mA-FSM** inicializou a sua estrutura considerando apenas uma regra nebulosa, sendo composta por duas regras nebulosas ao longo da segunda etapa de aprendizado. Deve ser observado que, embora o número de regras nebulosas se mantenha constante durante o ajuste do modelo, isso não implica que as regras que compõem a estrutura do modelo sejam as mesmas, já que enquanto uma regra é gerada outra pode estar sendo eliminada de forma simultânea.

O período de previsão para todos os modelos é composto por dados a partir de 1968-III

até 1993-IV, dando um total de 102 dados de teste ou validação.

A seguir, os modelos estimados foram avaliados, com o intuito de verificar a qualidade do ajuste. Neste trabalho, a verificação foi feita via uma análise dos resíduos estimados para o conjunto de treinamento. Assim, se o modelo for adequado, os resíduos estimados deverão ser aproximadamente não-correlacionados. Dois testes de adequação foram usados: o teste de autocorrelação residual e o teste de correlação cruzada entre os resíduos estimados e a série temporal.

Indicando por $\widehat{\rho}_k$ as autocorrelações estimadas e por \widehat{s}_k as correlações cruzadas estimadas, sendo k os atrasos, os resíduos estimados $\widehat{a}_t = \hat{y}^t - y^t$ são não-correlacionados se $\widehat{\rho}_k \sim 0$ e $\widehat{s}_k \sim 0$, para $k = 1, 2, \dots$. As autocorrelações e correlações cruzadas são dadas por:

$$\widehat{\rho}_k = \frac{\sum_{t=k+1}^n \widehat{a}_t \widehat{a}_{t-k}}{\sum_{t=1}^n \widehat{a}_t^2} \quad (7.6)$$

$$\widehat{s}_k = \frac{\sum_{t=k+1}^n \widehat{a}_t (y^{t-k} - \bar{y})}{\left[\sum_{t=1}^n \widehat{a}_t^2 \sum_{t=1}^n (y^t - \bar{y})^2 \right]^{1/2}} \quad (7.7)$$

sendo y^t o valor da série para o instante t , \bar{y} o valor médio e n o número de dados. Se os valores de $\widehat{\rho}_k$ e \widehat{s}_k são significativamente diferentes de zero, para um dado nível de significância α , os resíduos estimados são correlacionados. Caso contrário, os resíduos estimados são não-correlacionados. Para mais detalhes ver (Morettin e Tolo, 2004) e (de Holanda Sales et al., 1998).

A Figura 12 ilustra a análise dos resíduos para o modelo **C-FSM**, sendo que as linhas pontilhadas indicam os limites $\pm 2/\sqrt{n}$, os quais fornecem uma indicação geral de quebra do comportamento de ruído branco, com n igual ao número de graus de liberdade associado ao modelo. Ambos testes mostram que o modelo ajustado é adequado, pois os resíduos estimados são não correlacionados. A Figura 12-(a) e 12-(a) ilustram as funções de autocorrelação residual estimada do erro e correlação cruzada, respectivamente, entre a saída desejada e a saída estimada sob o conjunto de treinamento.

A Figura 13 ilustra a análise dos resíduos para o modelo **C-FSM_t**. A Figura 13-(a) ilustra a função de autocorrelação residual estimada, enquanto a função de correlação cruzada estimada é apresentada na Figura 13-(b).

Mesmo que as correlações cruzadas estimadas estejam dentro dos limites de confiança (Figura 13-(b)), verifica-se que existe informação não capturada pelo modelo **C-FSM_t**, o que o torna pouco adequado para a modelagem da série, quando comparado ao modelo **C-FSM**. Este comportamento é justificado, pois o modelo **C-FSM_t** foi construído sob a série não

estacionária.

Os coeficientes de autocorrelação e de correlação cruzada estimados para os modelos **A-FSM** e **mA-FSM** são apresentados nas 14 e 15, respectivamente, mostrando que os modelos obtidos no aprendizado sequencial foram bem ajustados, sendo adequados para a modelagem da série em estudo.

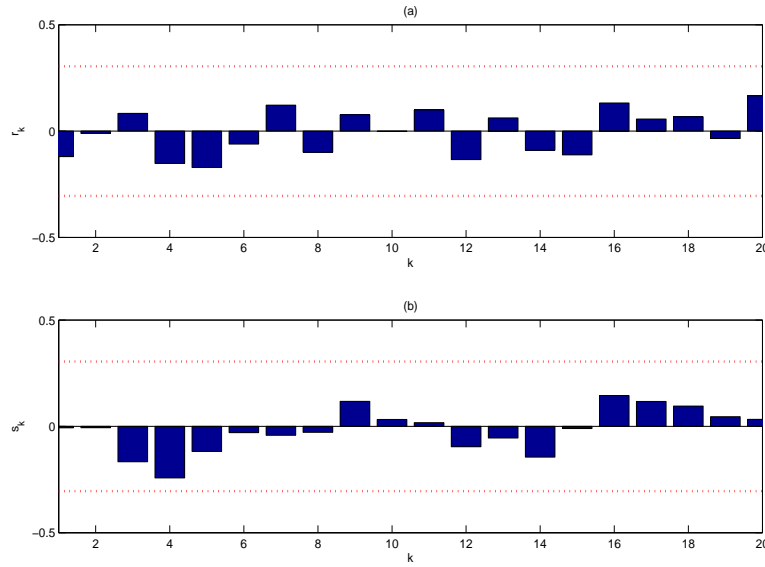


Figura 12: Análise dos resíduos para o modelo **C-FSM**: (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.

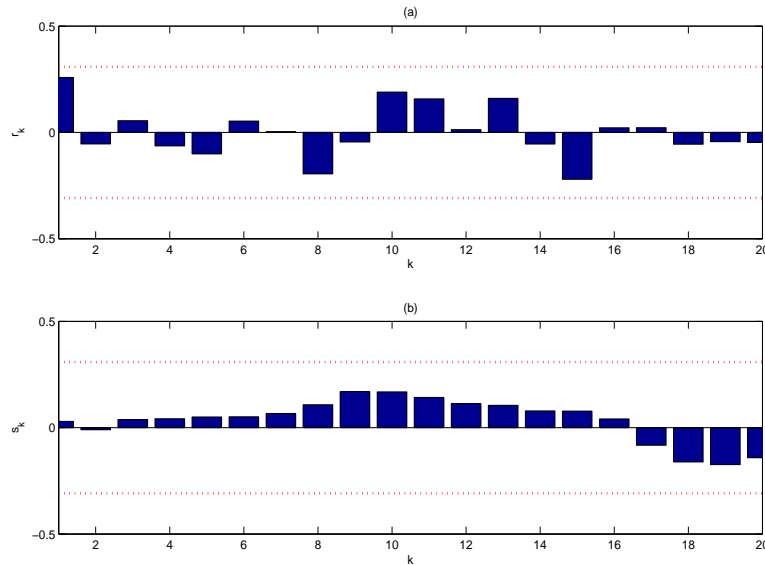


Figura 13: Análise dos resíduos para o modelo **C-FSM_t**: (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.

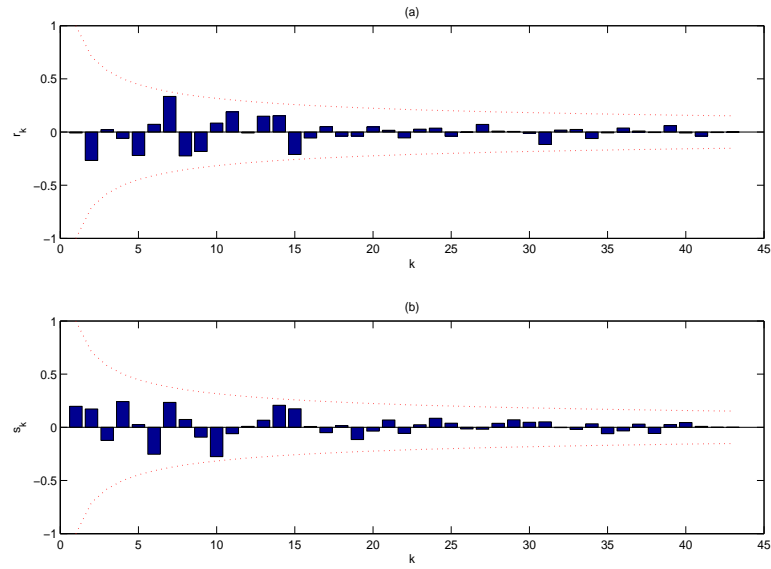


Figura 14: Análise dos resíduos para o modelo **A-FSM**: (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.

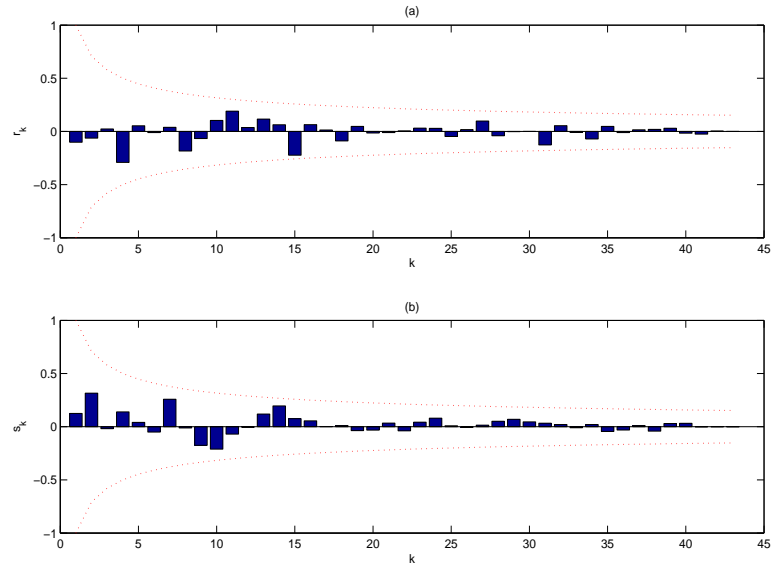


Figura 15: Análise dos resíduos para o modelo **mA-FSM**: (a) função de autocorrelação estimada, (b) função de correlação cruzada estimada.

Os modelos foram comparados utilizando o erro quadrático médio (**EQM**) como métrica de desempenho, na previsão de um até cinco passos à frente, a partir de 1968-III, já que esta medida de erro é a utilizada para a avaliação do desempenho dos modelos de séries temporais ajustados em (Tsay, 2005). A Tabela 3 apresenta os erros obtidos para cada modelo de

previsão e para os diferentes horizontes de previsão considerados.

Tabela 3: Desempenho dos modelos de previsão de um até cinco passos à frente para a série trimestral de desemprego dos Estados Unidos: erro quadrático médio (**EQM**).

Modelo	Horizonte de previsão				
	1	2	3	4	5
C-FSM_t	0,13	0,31	0,53	0,95	1,47
C-FSM	0,09	0,11	0,14	0,14	0,12
A-FSM	0,11	0,13	0,16	0,17	0,14
mA-FSM	0,09	0,13	0,14	0,16	0,14
Linear	0,08	0,31	0,67	1,13	1,54
MSA	0,10	0,43	0,94	1,64	2,48
TAR	0,08	0,32	0,66	1,11	1,59

Da Tabela 3, observa-se que o desempenho dos modelos **C-FSM**, **A-FSM** e **mA-FSM** é similar ao modelo linear, TAR e MSA no caso da previsão um passo à frente. No entanto, verifica-se que o desempenho dos modelos de séries temporais é deteriorado na previsão vários passos à frente.

Já os modelos propostos neste trabalho fornecem resultados de previsão acurados tanto na previsão um passo à frente como na previsão até cinco passos à frente. No caso do modelo **C-FSM_t**, os erros de previsão foram similares aos obtidos pelos modelos ajustados em (Tsay, 2005), porém, o seu desempenho foi inferior aos outros modelos baseados em regras ajustados.

A Figura 16 ilustra os resultados de previsão até cinco passos a frente para os modelos **C-FSM** e **C-FSM_t**, respectivamente.

Comparando os resultados de simulação obtidos para ambos os modelos construtivos, observa-se que, neste caso, o pré-processamento da série influencia de forma notável o desempenho dos modelos de série temporal obtidos. Embora os modelos sejam ajustados para previsões um passo à frente, a obtenção de previsões eficientes até vários passos à frente mostra de forma empírica a adequação e robustez dos modelos de previsão obtidos.

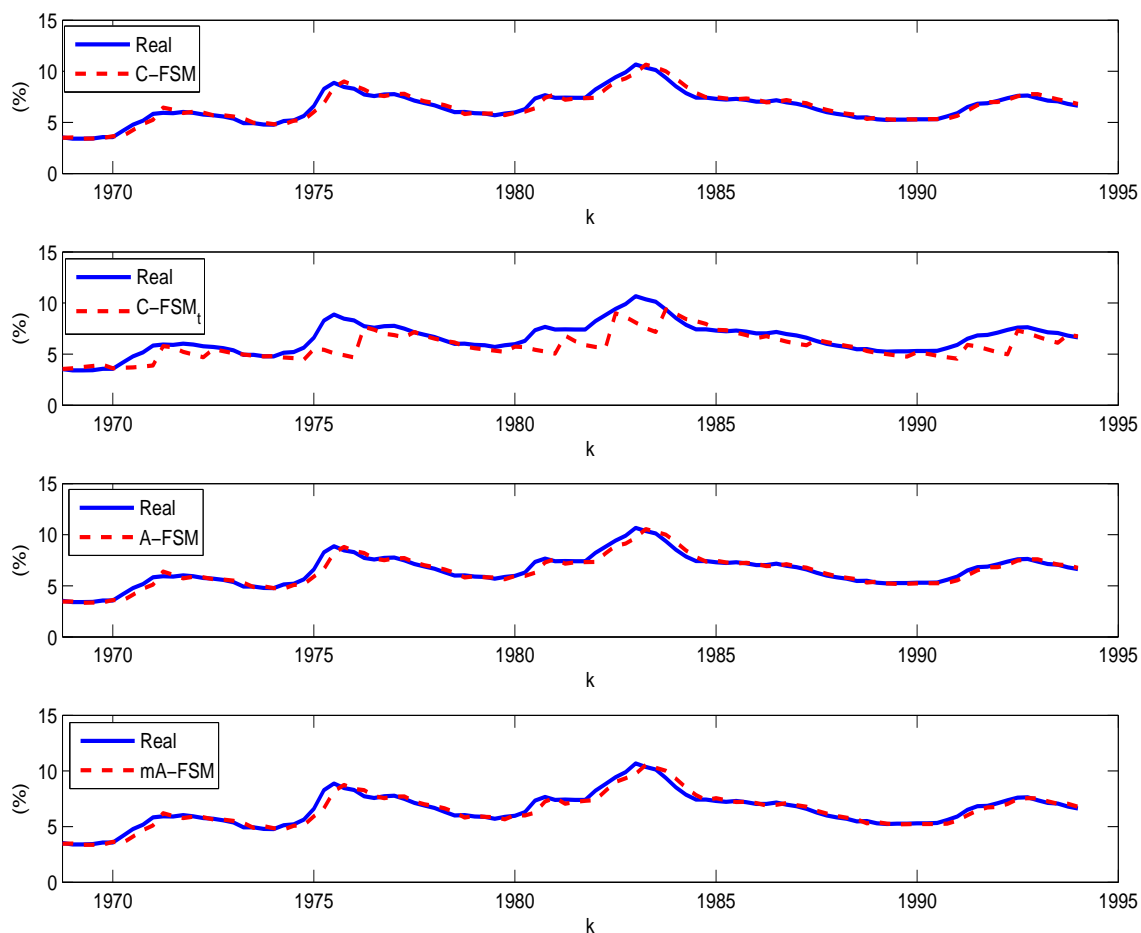


Figura 16: Previsão até 5 passos à frente dos modelos baseados em regras nebulosas da taxa de desemprego trimestral dos Estados Unidos.

7.4 Séries financeiras

As séries financeiras utilizadas neste trabalho fazem parte do conjunto reduzido de séries utilizado na Competição Neural de Previsão 2006/2007 (*NN3 Competition*), como parte do *International Joint Conference on Neural Networks'07*. As séries escolhidas são as séries NN3_102, NN3_104 e NN3_109 de um total de 11 séries temporais. Este conjunto de séries é composto por séries financeiras, assim como por séries industriais, cujos dados faltantes, assim como maiores informações e características serão disponibilizados em meados do ano de 2008.

As métricas de erro utilizadas para verificar o desempenho dos modelos foram o erro

relativo médio simétrico (**ERMs** (%)) e o erro absoluto médio (**EAM**). Estas métricas são calculadas utilizando as seguintes Equações:

$$ERM_s = \frac{100}{N} \sum_{k=1}^N \frac{|y^k - \widehat{y}^k|}{(y^k + \widehat{y}^k)/2} \quad (7.8)$$

$$EAM = \frac{1}{N} \sum_{k=1}^N |y^k - \widehat{y}^k| \quad (7.9)$$

onde, N é o número de padrões no conjunto de teste ou validação, \widehat{y}^k é a saída estimada pelo modelo de previsão e y^k é a saída desejada.

O **ERMs** é uma medida de erro empregada na Competição de Previsores com Redes Neurais (*Neural Networks Competition - NN3*). Utilizando esta medida, elimina-se o problema de obter erros grandes quando y^k é próximo de zero, e de diferenças muito grandes entre os erros percentuais absolutos quando y^k é maior que \widehat{y}^k e vice-versa (Makridakis e Hibon, 2000). Por exemplo, quando $y^k = 100$ e $\widehat{y}^k = 50$, o erro percentual médio é igual a **ERM**= 50%, enquanto que, para $y^k = 50$ e $\widehat{y}^k = 100$ o erro percentual médio é igual a **ERM**= 100%. Já no caso do **ERMs**, este será igual a 66,67% para ambos os casos. Além disso, o **ERM** é ilimitado, enquanto o **ERMs** adota valores no intervalo $[-200\%, 200\%]$.

A seguir são apresentados os resultados para as três séries NN3_102, NN3_104 e NN3_109.

7.4.1 Série NN3_102

A série NN3_102 é composta por um total de 126 observações mensais, de Janeiro de 1979 até Junho 1989. Esta série é composta por um componente sazonal e um componente aleatório, sendo necessário neste caso, retirar o componente sazonal para proceder com a seleção das variáveis de entrada para os modelos baseados em regras nebulosas.

Para a identificação do modelo de série temporal, utiliza-se apenas os dados considerados na etapa de aprendizado construtivo, no caso do modelo **C-FSM** e na etapa de inicialização *offline* no caso dos modelos com aprendizado sequencial **A-FSM** e **mA-FSM**. Os primeiros 108 dados do histórico da série são utilizados para o ajuste dos modelos, deixando os últimos dezoito dados para a etapa de validação ou teste dos modelos ajustados.

O conjunto inicial de possíveis entradas definido pelo algoritmo FNN para a série NN3_102, considerando os primeiros 9 anos que compõem o conjunto de treinamento é composto pelos

primeiros 9 atrasos da série. Com base neste conjunto, as entradas finais determinadas pelo critério PMI para o modelo construtivo são os atrasos 1 e 3, como mostra a Tabela 4. Como pode ser observado, o processo foi finalizado na terceira iteração, pois o valor de PMI obtido para o atraso 9 (uma possível entrada) foi inferior ao limiar definido para o critério PMI nessa iteração. Os valores de PMI obtidos para as entradas selecionadas a cada iteração são mostrados na terceira coluna da Tabela 4, enquanto os limiares de PMI respectivos são mostrados na quarta coluna.

Tabela 4: Resultados do algoritmo FNN-PMI para a série NN3_102.

Iteração	NN3_102 (9 anos)			NN3_102 (4 anos)		
	Lag	PMI	Limiar	Lag	PMI	Limiar
1	1	0,7775	0,1585	1	0,6405	0,2389
2	3	0,2348	0,1010	2	0,2166	0,1215
3	9	0,0744	0,0956	4	0,0465	0,1075

Ainda na Tabela 4 são mostradas as entradas selecionadas para o ajuste dos modelos sequenciais (**A-FSM** e **mA-FSM**). Neste caso, os atrasos são distintos pois considerou-se um conjunto de dados, para a seleção das entradas composto pelos primeiros quatro anos do histórico da série.

Como mostra a quinta coluna da Tabela 4, os atrasos determinados após a aplicação do algoritmo FNN-PMI são 1 e 2. A sexta e sétima coluna da Tabela 4 mostram os PMIs e limiares obtidos a cada iteração para os modelos com aprendizado sequencial.

Esta diferença nas variáveis finais se deve ao número reduzido de dados utilizados pelo algoritmo FNN assim como, para calcular as estimativas do PMI, principalmente para o caso dos modelos sequenciais. Deve-se observar que, quanto maior o número de dados utilizados para o cálculo destas estimativas, melhores serão as aproximações.

A estrutura do modelo construtivo é inicializada, considerando os critérios e algoritmos definidos no Capítulo 5. Durante a etapa de aprendizado, considerou-se $\alpha_{min} = 0,025$, sendo o processo finalizado após 40 épocas, obtendo uma estrutura final composta por três regras nebulosas.

O algoritmo SC utilizado na fase de inicialização do **mA-FSM** considerou $r_a = 0,75$ e $r_b = 1,75r_a$. Após esta etapa, procedeu-se o ajuste sequencial sob os próximos valores do conjunto de treinamento. Em ambos os casos (**A-FSM** e **mA-FSM**), considerou-se $\alpha_{min} = 0,01$, $f_{forget} = 0,95$, $T = 20$ e $\epsilon = 0,50$. Os modelos sequenciais finalizaram o

aprendizado com estruturas compostas por duas regras nebulosas em ambos os casos, embora o modelo **mA-FSM** tenha oscilações quanto ao número de regras durante o aprendizado tal que, $M^k \in [2, 3]$.

A Tabela 5 apresenta os erros obtidos na previsão um passo à frente no conjunto de treinamento e de validação, assim como o número de regras nebulosas que compõem a estrutura dos modelos de séries temporais no final do processo de aprendizado. A Tabela 5 também apresenta os erros de previsão de um até dezoito passos à frente obtidos para o conjunto de teste ou validação. Embora o aprendizado sequencial permita a adaptação do modelo com a apresentação de um novo dado, neste caso, o aprendizado foi finalizado antes de efetuar o teste, tal que o mesmo modelo seja utilizado para o cálculo das previsões um e múltiplos passos à frente.

Tabela 5: Erros de previsão para a série NN3_102.

Modelo	Treinamento 1 passo à frente			Validação 1 passo à frente		Validação Até 18 passos à frente	
	M	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)
C-FSM	3	3,41	179,36	4,72	287,98	11,40	658,05
A-FSM	2	3,23	175,92	4,62	303,13	11,58	695,77
mA-FSM	2	4,36	225,47	2,68	183,16	11,51	695,68

Como era de se esperar, o desempenho dos modelos deteriora a medida que o horizonte de previsão se estende, devido a realimentação do erro das previsões anteriores. Deve-se considerar que a função objetivo utilizada para o ajuste dos modelos utiliza apenas os erros na previsão um passo à frente.

A modo ilustrativo, a Figura 17 apresenta a variação do **ERMs** para diferentes horizontes de previsão, de 1 até dezoito passos à frente, para os três modelos ajustados. Como pode ser observado, embora o modelo **mA-FSM** tenha fornecido melhores resultados na previsão um passo à frente, os três modelos conseguiram resultados similares na previsão até dezoito passos à frente.

Os resultados de previsão um e até dezoito passos à frente para o modelo **C-FSM** são ilustrados na Figura 18-(a). A Figura 18-(b) e 18-(c) apresentam os resultados de previsão para os modelos **A-FSM** e **mA-FSM**, respectivamente.

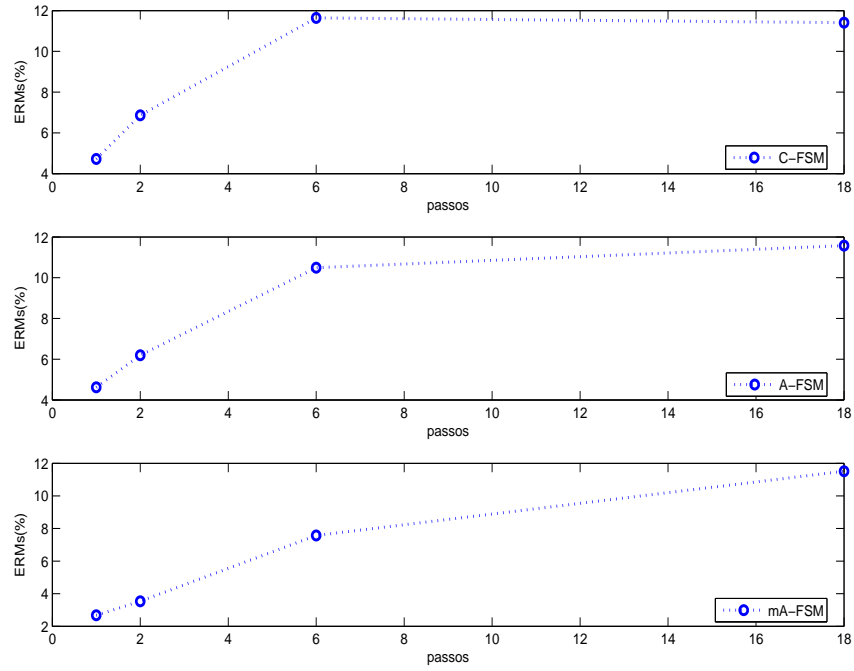


Figura 17: Variação do **ERMs** de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_102.

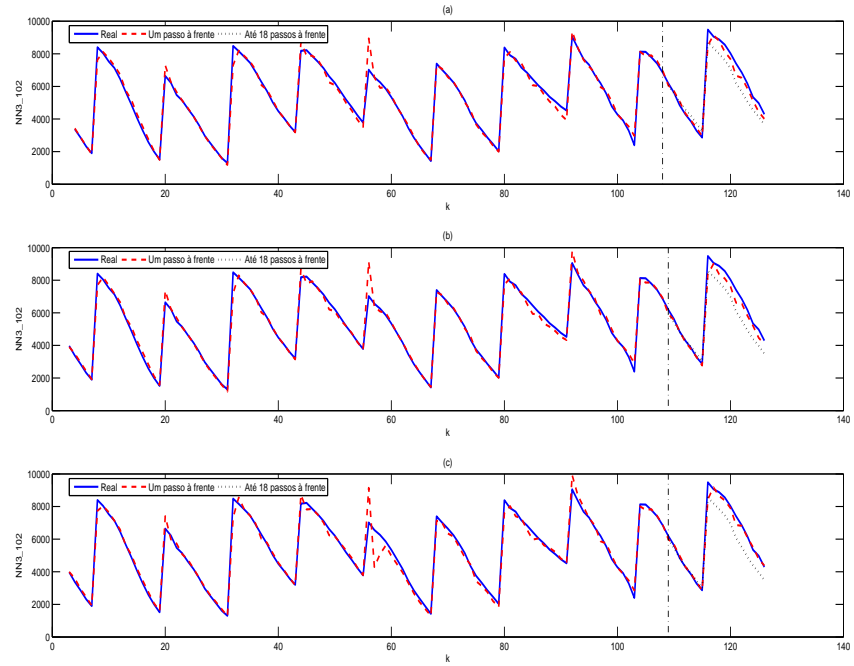


Figura 18: Previsão vários passos à frente para a série NN3_102: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

7.4.2 Série NN3_104

A série NN3_104 é composta por 115 observações mensais, no período de Janeiro de 1979 a Julho de 1988. Da mesma forma que a série NN3_102, a série NN3_104 é composta por um componente aleatório e um componente sazonal, tendo neste caso, que transformar a série original para obter a série estacionária.

Para a identificação do modelo construtivo, que implica na seleção de entradas e ajuste do modelo, utiliza-se os primeiros oito anos do histórico da série, deixando apenas os últimos dezoito valores para avaliar o desempenho do modelo.

O conjunto inicial de possíveis entradas para o modelo é definido pelo algoritmo FNN, sendo composto pelos primeiros 4 atrasos da série temporal. Após a construção deste conjunto inicial, o critério PMI é verificado para cada uma das possíveis variáveis de entrada. No final do processo iterativo, o conjunto final de entradas é composto pelos atrasos 1, 2 e 3, como mostra a Tabela 6. A primeira coluna mostra o número da iteração do processo de seleção de entradas utilizando o critério PMI e a segunda coluna mostra o atraso ou variável de entrada escolhido nessa iteração. A terceira coluna mostra o valor de PMI obtido pelo atraso selecionado, que corresponde ao máximo valor de PMI obtido pelas variáveis avaliadas nessa iteração. A quarta coluna, mostra o limiar para o PMI o qual deve ser inferior ao respectivo valor de PMI para que a entrada selecionada nessa iteração seja considerada como entrada relevante para a modelagem da série temporal. Como na quarta iteração esta condição não foi satisfeita, o processo é finalizado, considerando apenas as variáveis selecionadas nas três primeiras iterações como entradas para o modelo a ser ajustado.

As próximas colunas (quinta, sexta e sétima), mostram os resultados obtidos na seleção de entradas para os modelos com aprendizado sequencial, ou seja, considerando apenas os quatro primeiros anos do histórico da série na seleção de variáveis de entrada. Este mesmo conjunto de dados será utilizado na etapa de inicialização dos modelos **A-FSM** e **mA-FSM**, prévia ao processo de aprendizado dinâmico.

Como a Tabela 6 ilustra, apenas duas variáveis foram selecionadas para a construção dos modelos adaptativos (quinta coluna). Assim, os atrasos 1 e 3 da série temporal serão utilizados como variáveis de entrada para os modelos de previsão. Estas variáveis foram selecionadas a partir de um conjunto inicial composto pelos primeiros três atrasos da série temporal, sendo determinado pelo algoritmo FNN sob o conjunto de dados previamente definido.

Tabela 6: Resultados do algoritmo FNN-PMI para a série NN3_104.

	NN3_104 (8 anos)			NN3_104 (4 anos)		
Iteração	Lag	PMI	Limiar	Lag	PMI	Limiar
1	1	0,3793	0,1411	1	0,3989	0,2332
2	3	0,1705	0,1058	3	0,2314	0,1550
3	2	0,0351	0,0303	2	0,0293	0,0445
4	4	0,0249	0,0322	—	—	—

Uma vez que as entradas dos modelos são definidas, os modelos construtivo e *online* são inicializados e ajustados.

Como já foi mencionado, o modelo **C-FSM** utiliza os primeiros 9 anos do histórico da série durante a etapa de inicialização da estrutura, assim como durante o aprendizado construtivo descrito no Capítulo 5. O valor de α_{min} considerado para ambas as etapas foi igual a 0,025 e o processo foi finalizado após 40 épocas, sendo a estrutura final composta por $M = 3$ regras nebulosas. Já os modelos **A-FSM** e **mA-FSM** utilizam apenas os primeiros quatro anos do histórico da série para a inicialização da estrutura, e os próximos quatro anos da série para a etapa de ajuste sequencial.

O modelo **mA-FSM** inicializou a sua estrutura com $M = 2$ regras nebulosas, considerando valores de $r_a = 0,50$ e $r_b = 1,50r_a$ para o algoritmo SC. Ambos os modelos utilizaram os valores de $\alpha_{min} = 0,01$, $f_{forget} = 0,95$, $T = 20$ e $\epsilon = 0,50$ para o aprendizado *online*. O modelo **A-FSM** é composto por $M = 3$ regras nebulosas no final do aprendizado. Os modelos obtidos no final da etapa de aprendizado são utilizados para estimar os próximos dezoito valores do histórico da série, nas previsões um passo à frente e até dezoito passos à frente.

A Tabela 7 apresenta o número final de regras nebulosas que compõe a estrutura dos modelos baseados em regras nebulosas no final da etapa de aprendizado. A Tabela 7 também apresenta os erros de previsão obtidos para o conjunto de treinamento e na etapa de validação dos modelos, na previsão um e até dezoito passos à frente.

Em termos gerais, os três modelos conseguiram resultados de previsão bastante próximos quando comparados entre si, com estruturas compactas apesar de cada uma utilizar um algoritmo de aprendizado diferente. Como pode ser observado na Tabela 9, os valores mais baixos de **ERMs** e **EAM** obtidos na previsão um passo à frente correspondem aos resultados obtidos pelo modelo construtivo. Porém, os erros percentuais simétricos e absolutos médios

obtidos na previsão até dezoito passos à frente, são os obtidos pelos modelos *online* **A-FSM** e **mA-FSM**.

Tabela 7: Erros de previsão para a série NN3_104.

Modelo	Treinamento 1 passo à frente			Validação 1 passo à frente		Validação Até 18 passos à frente	
	M	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)
C-FSM	3	10,98	438,27	6,75	334,93	12,32	593,39
A-FSM	3	11,98	487,60	7,40	348,35	11,91	578,27
mA-FSM	2	10,56	452,42	7,96	386,55	11,74	573,57

A Figura 19 ilustra a variação do **ERMs** para diferentes horizontes de previsão. Neste caso, a variação do erro à medida que o horizonte de previsão aumenta é similar para os três modelos de previsão.

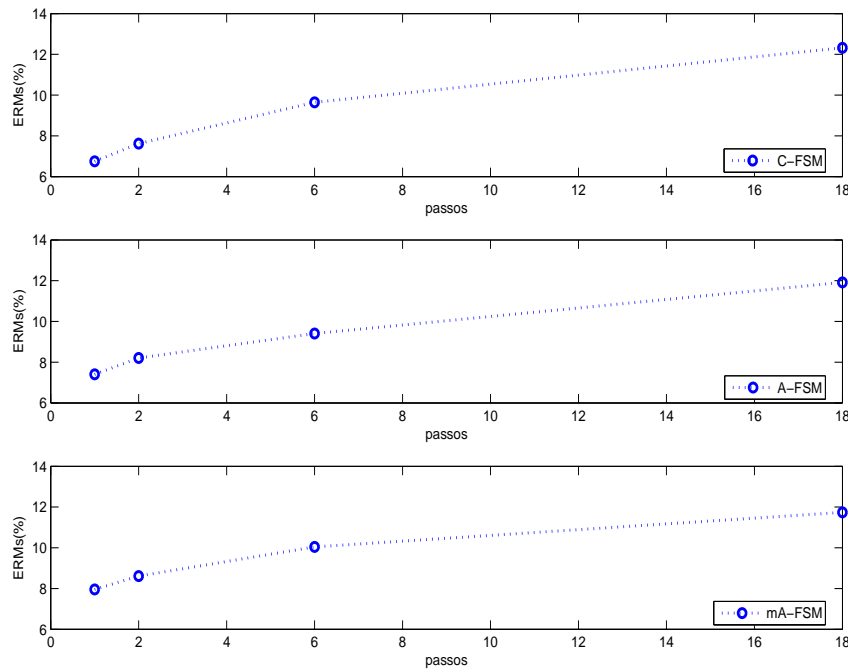


Figura 19: Variação do **ERMs** de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_104.

Finalmente, os resultados de previsão, um e até dezoito passos à frente são ilustrados na

Figura 20, para cada um dos modelos ajustados.

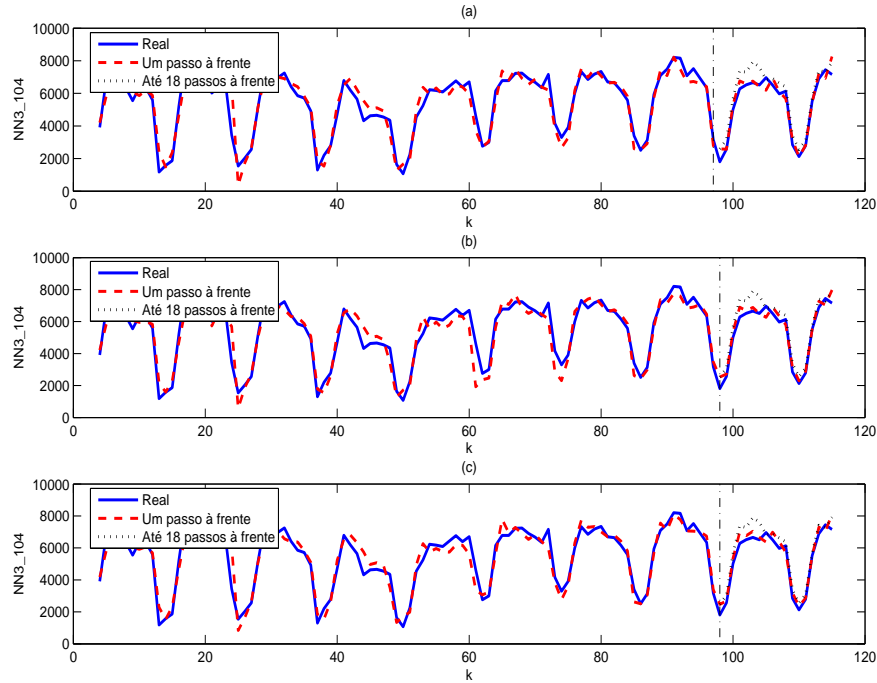


Figura 20: Previsão vários passos à frente para a série NN3_104: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

7.4.3 Série NN3_109

A série NN3_109 é composta por 123 observações mensais, no período de Janeiro de 1979 a Março de 1989. Neste caso, diferente das duas séries anteriormente apresentadas, a série é definida como uma série não sazonal, sendo composta por um componente de tendência de primeira ordem e o componente aleatório. A série foi transformada, trabalhando assim, como a primeira diferença da série.

Para a identificação do modelo construtivo, utiliza-se os dados do histórico da série até Setembro de 1987, dando um total de 105 observações e deixando apenas os últimos dezoito valores para avaliar o desempenho do modelo.

O conjunto inicial de possíveis entradas para o modelo é definido pelo algoritmo FNN, sendo composto pelos primeiros 6 atrasos da série temporal. Após a construção deste conjunto inicial, o critério PMI é verificado para cada uma das possíveis variáveis de entrada. No final do processo iterativo, o conjunto final de entradas é composto pelos atrasos 2, 3 e 6, como mostra a Tabela 8. A primeira coluna mostra o número da iteração do processo de seleção

de entradas utilizando o critério PMI, a segunda coluna mostra o atraso ou variável de entrada escolhido nessa iteração. A terceira coluna mostra o valor de PMI obtido pelo atraso selecionado, que corresponde ao máximo valor de PMI obtido pelas variáveis avaliadas nessa iteração. A quarta coluna, mostra o limiar do PMI estimado para cada iteração do processo de seleção de entradas, o qual, como pode ser visto, finaliza quando o valor de PMI obtido na iteração final é inferior ao limiar calculado.

As próximas colunas (quinta, sexta e sétima), mostram os resultados obtidos na seleção de entradas para os modelos com aprendizado sequencial, ou seja, considerando apenas os quatro primeiros anos do histórico da série na seleção de variáveis de entrada. Este mesmo conjunto de dados, seguindo o procedimento geral, será utilizado na etapa de inicialização dos modelos **A-FSM** e **mA-FSM**.

Para esta série, como a Tabela 8 ilustra, a única variável de entrada a ser considerada para a construção dos modelos adaptativos é o atraso 3. Esta variável foi selecionada a partir de um conjunto inicial composto pelos primeiros quatro atrasos da série temporal, sendo este número de atrasos determinado pelo algoritmo FNN sob o conjunto de dados entrada-saída construído utilizando as observações dos primeiros quatro anos do histórico da série.

Tabela 8: Resultados do algoritmo FNN-PMI para a série NN3_109.

	NN3_105 (aprox. 9 anos)			NN3_105 (4 anos)		
Iteração	Lag	PMI	Limiar	Lag	PMI	Limiar
1	3	0,3037	0,2285	3	0,3449	0,3005
2	2	0,2030	0,1995	2	0,2024	0,2517
3	6	0,1418	0,1169	—	—	—
4	1	0,0609	0,0783	—	—	—

Uma vez que as entradas dos modelos são definidas, os modelos construtivo e *online* são inicializados e ajustados.

Como já foi mencionado, o modelo **C-FSM** utiliza os primeiros 9 anos do histórico da série durante a etapa de inicialização da estrutura, assim como durante o aprendizado construtivo descrito no Capítulo 5. O valor de α_{min} considerado para ambas as etapas foi igual a 0,01 e o processo foi finalizado após 40 épocas, sendo a estrutura final composta por $M = 4$ regras nebulosas.

Já os modelos **A-FSM** e **mA-FSM** utilizam apenas os primeiros quatro anos do histórico

da série durante a etapa de inicialização da estrutura, e os próximos quatro anos da série para a etapa de ajuste sequencial.

O modelo **mA-FSM** é composto por apenas $M = 1$ regra nebulosa, considerando valores de $r_a = 0,50$ e $r_b = 1,50r_a$ para o algoritmo SC. Tanto o modelo **mA-FSM** como o modelo **A-FSM** utilizaram os valores de $\alpha_{min} = 0,01$, $f_{forget} = 0,95$, $T = 20$ e $\epsilon = 0,50$ para o aprendizado *online*. O modelo **A-FSM** é composto por $M = 2$ regras nebulosas no final do aprendizado.

O número de regras nebulosas para cada modelo ajustado é resumido na Tabela 9. A Tabela 9 também apresenta os erros de previsão obtidos para o conjunto de treinamento e na etapa de validação dos modelos, na previsão um passo à frente e até dezoito passos à frente.

Em geral, os três modelos conseguiram resultados de previsão bastante próximos quando comparados entre si. Porém, o modelo **mA-FSM** obteve resultados similares com apenas uma regra nebulosa. Ou seja, o modelo obtido neste caso, equivale a um único modelo de regressão, com parâmetros ajustados utilizando o algoritmo EM sequencial, utilizando como variáveis as entradas selecionadas via o algoritmo FNN-PMI.

Tabela 9: Erros de previsão para a série NN3_109.

Modelo	Treinamento 1 passo à frente			Validação 1 passo à frente		Validação Até 18 passos à frente	
	M	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)
C-FSM	4	3,68	133,95	3,41	116,49	3,74	128,41
A-FSM	2	4,15	54,74	3,48	121,79	3,57	125,28
mA-FSM	1	2,08	70,18	3,66	125,68	3,71	128,05

A Figura 21 ilustra a variação do **ERMs** para diferentes horizontes de previsão. Os três modelos atingiram patamares similares de erros relativos simétricos, mostrando o modelo **mA-FSM** como o mais estável, já que o **ERMs** mantém-se praticamente constante para diferentes horizonte de previsão.

Os resultados de previsão, um e até dezoito passos à frente são ilustrados na Figura 22, para cada um dos modelos ajustados.

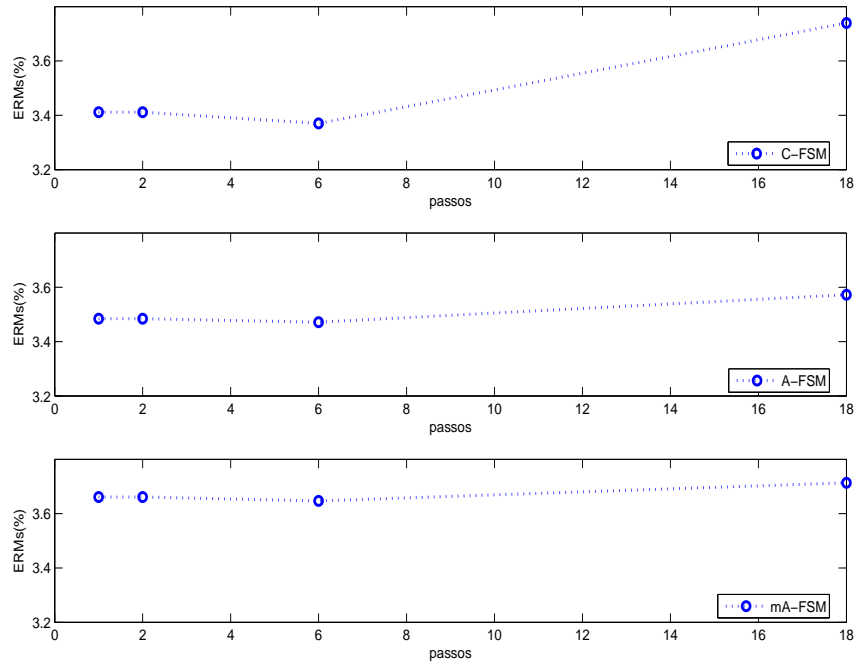


Figura 21: Variação do **ERMs** de um até dezoito passos à frente para os modelos de série temporal obtidos, série NN3_109.

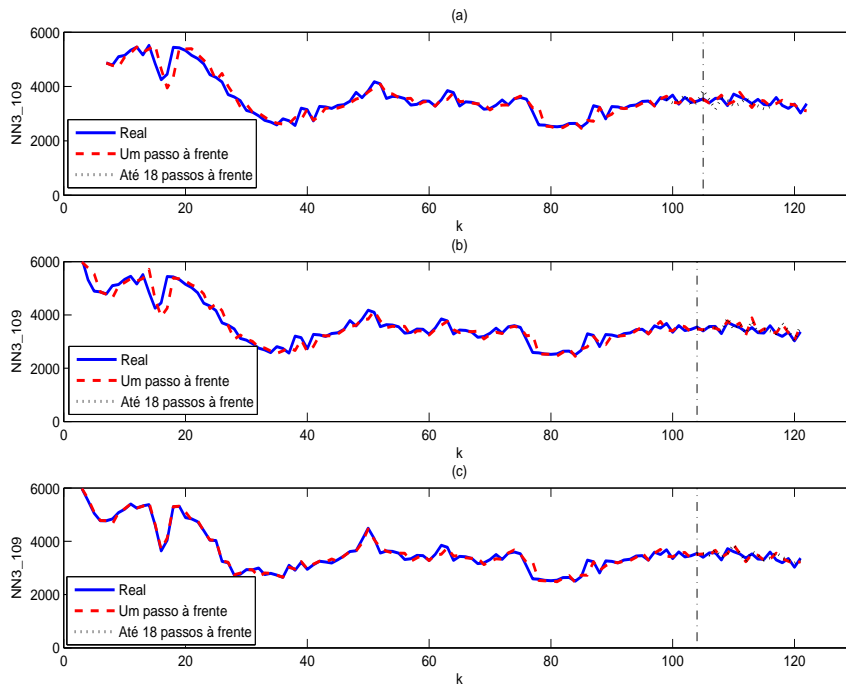


Figura 22: Previsão vários passos à frente para a série NN3_109: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

7.5 Série de vazões mensais

O planejamento e operação dos sistemas de energia e recursos hídricos envolve características de produção não-lineares e requer a utilização de inúmeras variáveis de entrada, sendo uma delas a previsão de vazões, necessárias para o propósito de simulação, otimização e tomada de decisões. Por este motivo, as previsões ao longo do período de planejamento devem ser as mais acuradas possíveis.

As séries de vazões possuem componentes sazonais, motivo pelo qual o setor elétrico brasileiro desagrega a modelagem de cada série de vazão, tal que doze modelos independentes sejam ajustados para cada série. No entanto, esta abordagem possui um alto custo computacional, já que são mais de cem séries de vazões que necessitam ser modeladas.

Além de realizar testes de simulação considerando esta abordagem, este trabalho também apresenta resultados de simulação ajustando apenas um modelo baseado em regras para a série temporal.

A série em estudo corresponde ao histórico de vazões afluentes mensais da usina hidroelétrica de Furnas, localizada na região sudeste do Brasil, sendo o histórico composto por dados a partir do ano de 1931 até o ano de 2005.

Para esta aplicação, o processo de modelagem segue a seqüência de passos descrita no Capítulo 1. Assim, o processo é inicializado com o pré-processamento dos dados e a seleção de variáveis de entrada para o modelo de série temporal.

Um conjunto de entradas é selecionado aplicando a abordagem combinada FNN-PMI. Para a aplicação deste algoritmo, a série de vazão é transformada para a sua versão estacionária, usando a Equação (2.1) do Capítulo 2.

O conjunto de dados utilizados nesta fase de análise e seleção de entradas são os mesmos utilizados na etapa de inicialização dos modelos baseados em regras. Dois períodos de teste são apresentados. O primeiro corresponde ao histórico da série de Janeiro de 1981 a Dezembro de 1985. Este período é composto pelo ano de 1983 o qual é considerado como um ano úmido, já que a média de vazão mensal é superior a média histórica para todos os meses desse ano. Neste caso, a seleção de entradas para o modelo construtivo é realizada considerando os primeiros 50 anos do histórico da série, ou seja, dados a partir de Janeiro de 1931 até Dezembro de 1980.

O segundo período de teste é composto pelos últimos dez anos do histórico da série. Ou seja, o período de validação dos modelos inicia em Janeiro de 1996 e finaliza em Dezembro

de 2005. Neste caso, a seleção de entradas é realizada considerando os primeiros 65 anos do histórico da série temporal.

No caso do modelo com aprendizado sequencial **A-FSM**, a seleção de entradas é realizada utilizando apenas o histórico de vazão de Janeiro de 1931 a Dezembro de 1950 resultando em um total de 20 anos para ambos os períodos de validação (de Janeiro de 1981 a Dezembro de 1985 e de Janeiro de 1996 a Dezembro de 2005).

O algoritmo FNN é utilizado para definir um conjunto inicial de possíveis entradas. Logo, o critério PMI é aplicado para determinar as entradas mais relevantes a partir do conjunto inicial definido previamente pelo algoritmo FNN.

Aplicando o algoritmo FNN, o conjunto inicial de possíveis entradas para o modelo **C-FSM** considerando os primeiros 65 anos foi composto pelos 6 primeiros atrasos da variável dependente, ou seja, da série de vazões. Por exemplo, para uma saída desejada y_k , o primeiro atraso (lag 1) é representado por y^{k-1} , o segundo atraso (lag 2) é representado por y^{k-2} , e assim por diante. A variação do número de falsos vizinhos a cada iteração do algoritmo FNN é ilustrado na Figura 23-(a). No caso do modelo **A-FSM** (20 primeiros anos) e do modelo **C-FSM** (50 primeiros anos), o conjunto inicial de possíveis entradas é composto pelos 5 primeiros atrasos, como ilustra a Figura 23-(b) e (c), respectivamente.

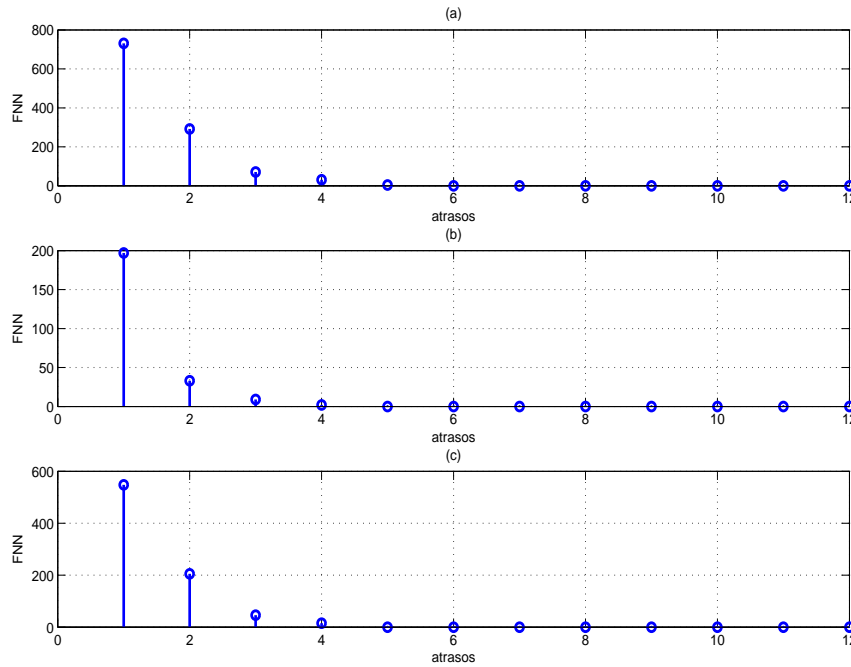


Figura 23: Definição do conjunto inicial de possíveis entradas via o algoritmo FNN, para a série mensal do posto de Furnas: (a) modelo **C-FSM** (65 anos), (b) modelo **A-FSM** (20 anos), (c) modelo **C-FSM** (50 anos).

Os resultados obtidos aplicando o critério PMI para ambos os modelos são apresentados na Tabela 10. Os atrasos mostrados nesta Tabela, são os que obtiveram maior valor de PMI.

Tabela 10: Resultados aplicando o algoritmo FNN-PMI para a série de vazões mensais do posto de Furnas.

C-FSM(50 anos)			C-FSM (65 anos)			A-FSM (20 anos)		
Lag	PMI	Limiar	Lag	PMI	Limiar	Lag	PMI	Limiar
1	0,4571	0,0820	1	0,4415	0,0703	1	0,4355	0,1223
4	0,1974	0,0620	4	0,1941	0,0547	4	0,2259	0,0898
2	0,0553	0,0305	2	0,0319	0,0319	2	0,0653	0,0474
5	0,0159	0,0311	6	0,0263	0,0357	3	0,0141	0,0272

A primeira, quarta e sétima coluna correspondem aos atrasos ou lags com maior PMI a cada iteração para cada um dos modelos ajustados. A segunda, quinta e oitava coluna mostram os respectivos valores de PMI obtidos para cada lag escolhido. Nesta Tabela, novamente pode-se observar que, a medida que as entradas vão sendo escolhidas, os valores de PMI para as próximas entradas são sempre menores, ou seja, o valor de PMI vai diminuindo rapidamente após uma entrada ter sido selecionada.

Assim, o valor de PMI obtido pela primeira entrada selecionada (a de maior relevância) será sempre maior que o valor de PMI obtido pelas seguintes. A terceira, sexta e nona coluna da Tabela 10 definem o limiar para o PMI a cada iteração, o qual é definido por *bootstrapping* (Luna et al., 2006a). Ou seja, se o PMI obtido pela entrada selecionada em uma determinada iteração for menor que o limiar para o PMI nessa mesma iteração, então, esta entrada não possui mais informação relevante para o sistema. Como o valor de PMI da entrada selecionada a cada iteração sempre diminui, então, todas as entradas ainda não selecionadas a partir dessa iteração serão descartadas e o processo de seleção é finalizado.

Pode-se concluir que apenas os atrasos 1, 2 e 4 são necessários para construir um modelo de série temporal para a série de vazões de Furnas. Particularmente, neste caso, ambos os modelos **C-FSM** e **A-FSM** para os conjuntos de dados diferentes possuem as mesmas variáveis de entrada. No entanto, deve-se observar que, o conjunto de variáveis de entradas é influenciado pelo número de dados utilizados no processo, pois as estimativas para o cálculo do PMI é sensível ao número de dados e, quanto maior for o número de dados utilizados, melhores serão as estimativas das funções de probabilidade marginal, condicional e inclusive dos valores esperados necessários.

Além disso, a janela de previsão definida pelo algoritmo FNN é também dependente do

número de dados utilizados pelo algoritmo. As entradas a serem utilizadas pelo modelo **mA-FSM** serão as mesmas utilizadas pelo modelo **A-FSM**. Assim, em termos gerais, os modelos de série temporal adotam a seguinte forma:

$$\hat{y}^k = f[y^{k-1}, y^{k-2}, y^{k-4}]$$

Uma vez que as entradas para o modelo de série temporal são selecionadas, a série estacionária é normalizada, de tal forma que todos os valores se encontrem no intervalo $[0, 1]$. Os dados de entrada-saída são construídos e, finalmente, procede-se à construção do modelo de previsão, a determinação da sua estrutura, assim como o ajuste dos parâmetros, que no caso das duas propostas deste trabalho (**C-FSM** e **A-FSM**), serão feitos de maneira simultânea e automática.

Como foi descrito no Capítulo 5, a construção do **C-FSM** requer uma inicialização, a qual é composta por duas regras nebulosas sendo que estas regras são geradas e ajustadas por meio do algoritmo EM. Uma vez que a estrutura é inicializada, procede-se o aprendizado construtivo *offline*, mediante uma seqüência de passos EM, em combinação com critérios de adição de novas regras nebulosas, assim como critérios de eliminação ou poda.

O limiar de α_i para a eliminação ou poda de regras foi definido como sendo igual a $\alpha_{min} = 0,005$ para o modelo **C-FSM**, de tal forma que, toda regra nebulosa com $\alpha_i < \alpha_{min}$ é eliminada da base de regras atual, a cada iteração.

No caso do modelo construtivo ajustado para o primeiro período de teste (de 1981 a 1985), a estrutura final obtida para o modelo de série temporal é composta por $M = 24$ regras nebulosas. Já para o segundo período de teste (de 1996 a 2005), utilizando um conjunto maior de dados de treinamento, a estrutura final do modelo é composta por $M = 19$ regras nebulosas.

Este mesmo valor foi adotado para a etapa de inicialização dos modelos com aprendizado seqüencial. Os valores para os parâmetros durante o aprendizado seqüencial foram iguais a $T = 3$, $\alpha_{min} = 0,001$, $f_{forget} = 0,99$ e $\epsilon = 0,5$, sendo o mesmo modelo utilizado para efetuar a previsão em ambos os períodos de validação. O modelo **A-FSM** obteve uma estrutura inicial composta por $M^0 = 11$ regras nebulosas, finalizando com apenas $M = 2$ regras nebulosas.

No caso do modelo **mA-FSM**, o algoritmo SC definiu uma estrutura inicial composta por $M^0 = 3$ regras nebulosas para $r_a = 0,25$ e $r_b = 1,50r_a$, modificando a sua estrutura durante o aprendizado, sendo finalmente composta por $M = 2$ regras nebulosas para ambos os períodos de teste ou validação. Os parâmetros durante o aprendizado seqüencial foram os

mesmos utilizados pelo modelo **mA-FSM**.

A verificação e adequação dos modelos foi baseada na análise de resíduos. Os testes de autocorrelação e correlações cruzadas foram aplicados para a estrutura do sistema nebuloso baseado em regras, obtida no final do aprendizado construtivo (**C-FSM**), como uma maneira de avaliar o modelo de previsão (Hamilton, 1994), (Morettin e Toloi, 2004). Para tal fim, os primeiros 20 atrasos das correlações residuais foram calculados.

A Figura 24 ilustra os coeficientes de autocorrelação e correlação cruzada estimadas para **C-FSM** ajustado no período de 1931 a 1980. A Figura 24-(a) apresenta as autocorrelações estimadas e os limites superiores e inferiores iguais a $\pm 1/\sqrt{n}$, onde n é o número de parâmetros livres do modelo de previsão no final do treinamento, ou seja, $n = N_{tr} - N_p$, sendo N_{tr} o número de dados de treinamento e N_p o número de parâmetros do modelo no final do processo de aprendizado.

A Figura 24-(b) ilustra as correlações cruzadas estimadas para os primeiros 20 atrasos entre a saída desejada e os valores residuais estimados, assim como os limites superior e inferior $\pm 1/\sqrt{n}$. Os limites superior e inferior são representados por linhas pontilhadas e fornecem uma indicação geral de possível quebra do comportamento de ruído branco (Box et al., 1994). Em termos gerais, a análise feita indicou uma modelagem adequada da série. De forma semelhante, a Figura 26 ilustra a análise dos resíduos para o modelo construtivo ajustado no período de 1931 a 1995.

As Figuras 26 e 27 ilustram os coeficientes de autocorrelação e correlação cruzada estimadas para os modelos **A-FSM** e **mA-FSM**, respectivamente. O primeiro trecho composto pelos primeiros 400 coeficientes corresponde a análise do modelo antes do período de 1981. Após este período o ajuste e a análise dos resíduos continuam para assim, estimar os valores da série no período de 1996 a 2005.

É interessante observar que, de acordo com as Figuras 26 e 27, a medida que os modelos com aprendizado seqüencial vão tomando conhecimento de novas informações referentes a dinâmica da série a modelar, a adequação do modelo é otimizada, pois os coeficientes de autocorrelação e de correlação cruzada vão se localizando dentro do intervalo de confiança construído, para ambos os casos.

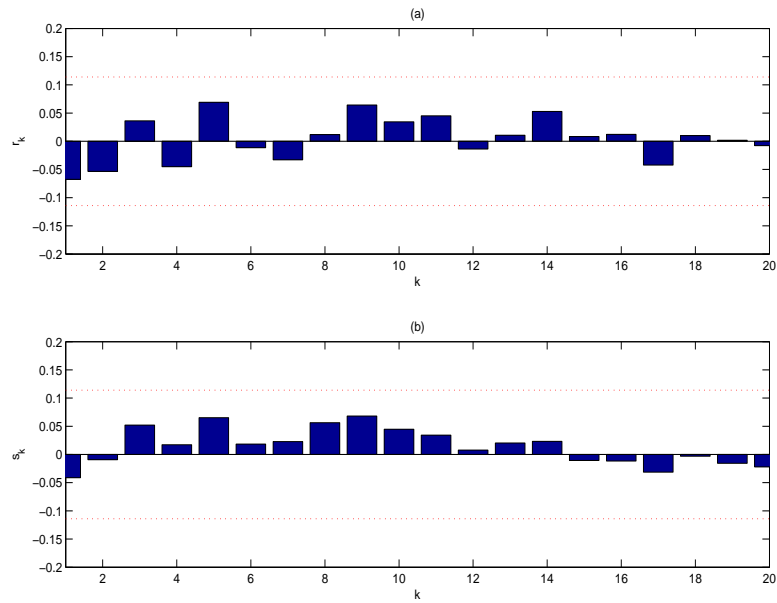


Figura 24: Análise dos resíduos para a série de Furnas utilizando o **C-FSM** na previsão um passo à frente, período de 1981 a 1985: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.

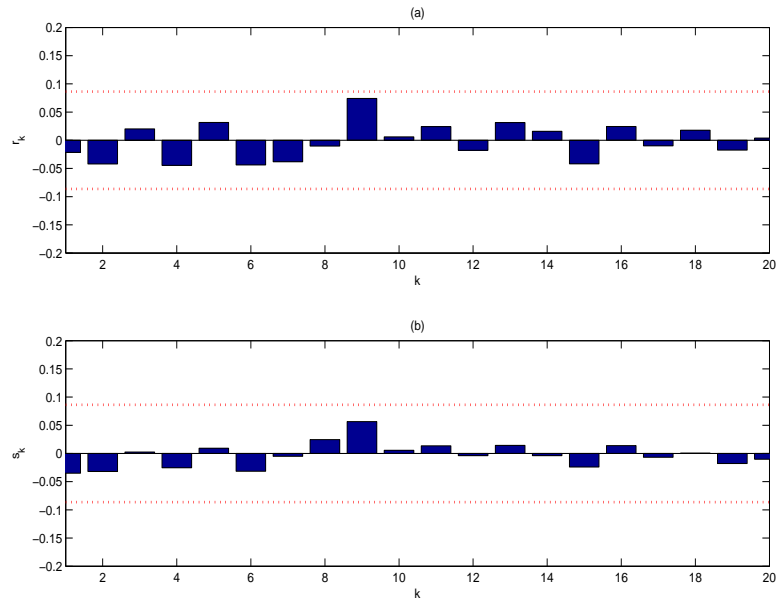


Figura 25: Análise dos resíduos para a série de Furnas utilizando o **C-FSM** na previsão um passo à frente, período de 1996 a 2005: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.

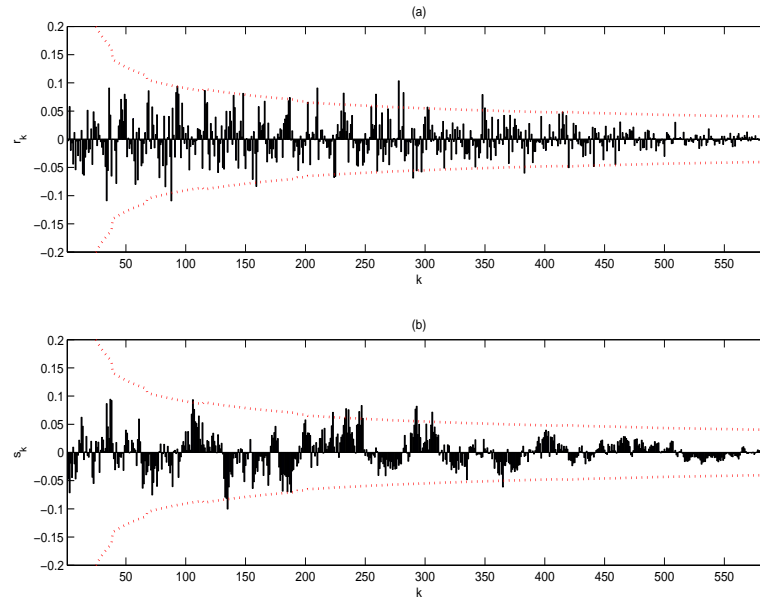


Figura 26: Análise dos resíduos para a série de Furnas utilizando o **A-FSM** na previsão um passo à frente: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.

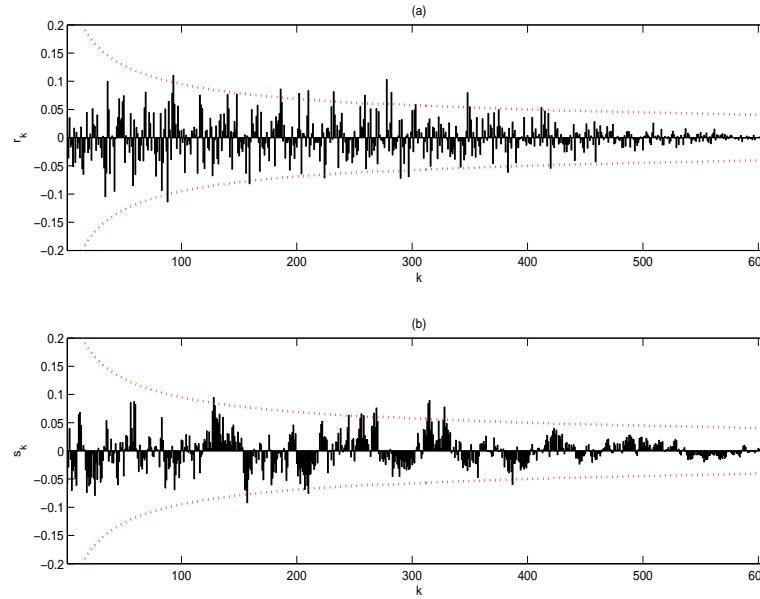


Figura 27: Análise dos resíduos para a série de Furnas utilizando o **mA-FSM** na previsão um passo à frente: (a) Função de autocorrelação estimada, (b) Função de correlação cruzada estimada.

A variação no tamanho da estrutura de cada modelo baseado em regras é ilustrado na Figura 28. A variação do número de regras para o modelo construtivo durante o aprendizado *offline* é ilustrado na Figura 28-(a) e (b), utilizando como conjunto de treinamento, os dados nos períodos de 1931 a 1980 e de 1931 a 1995, respectivamente. A atualização da estrutura do modelo **A-FSM** é apresentada na Figura 28-(c), onde observa-se que, o número de regras que compõem a estrutura do modelo varia entre 2 e 11, tendo a estrutura um maior número de regras nebulosas no início do aprendizado sequencial. Isto ocorre devido a que o número inicial de regras nebulosas é fornecida pela etapa de inicialização do modelo, a qual é realizada por meio de um aprendizado *offline*.

A Figura 28-(d) apresenta a variação da estrutura do modelo **mA-FSM**. Como pode ser observado, o modelo é composto por um máximo de 3 regras nebulosas e um mínimo de 2 regras nebulosas. Em geral, este modelo inicializa a sua estrutura com um número menor de regras nebulosas quando comparado ao modelo **A-FSM**. No entanto, deve-se observar que o número inicial de regras que compõem a estrutura do modelo depende de um conjunto de parâmetros, que no caso do algoritmo SC consiste principalmente nos parâmetros r_a e r_b . Além disso, embora o tamanho da base de regras se mantenha constante por uns períodos de tempo, deve-se observar que, isso não implica que não existam mudanças na estrutura do modelo, já que no mesmo instante, uma regra pode ser gerada, ao mesmo tempo que outra pode estar sendo eliminada da estrutura do modelo.

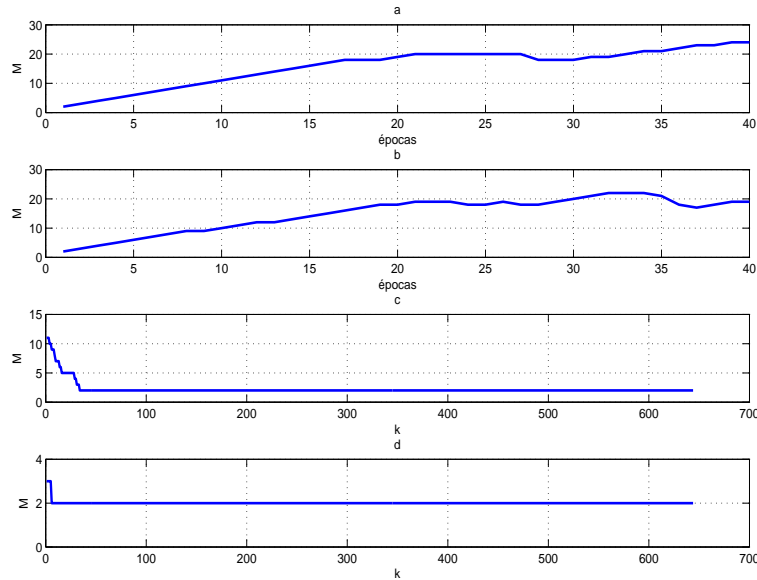


Figura 28: Variação de M para cada um dos modelos baseados em regras, ajustados para a série de vazões mensais do posto de Furnas: (a) **C-FSM** (período de 1981 a 1985), (b) **C-FSM** (período de 1996 a 2005), (c) **A-FSM**, (d) **mA-FSM**.

Os modelos obtidos são avaliados utilizando a raiz do erro quadrático médio (**REQM** (m^3/s)) definido pela Equação (7.2), erro relativo médio (**ERMs** (%)) calculado utilizando a Equação (7.8), erro absoluto médio (**EAM** (m^3/s)) dado pela Equação (7.9) e o índice de erro não-dimensional (**NDEI**) dado pela Equação 7.3.

Os resultados de simulação obtidos na previsão um passo à frente, para o período de Janeiro de 1981 a Dezembro de 1985 são ilustrados na Figura 29. Os resultados de previsão obtidos pelo modelo construtivo são apresentados na Figura 29-(a). A Figura 29-(b) e (c) ilustra as previsões obtidas pelos modelos **A-FSM** e **mA-FSM**, respectivamente.

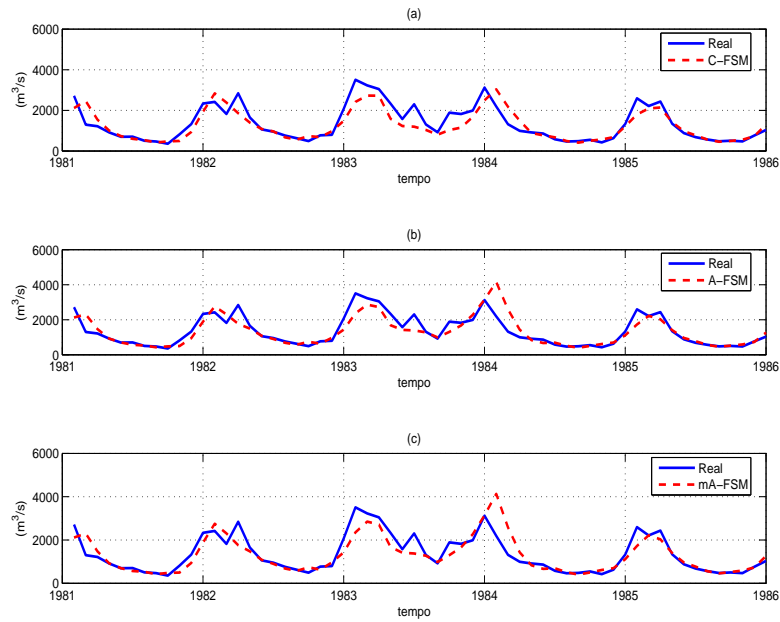


Figura 29: Resultados de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1981 a 1985: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

Os erros globais obtidos para este período do histórico de vazões mensais são apresentados na Tabela 11, para a previsão um passo à frente, mostrando também os resultados de simulação obtidos para o mesmo período utilizando a rede neural multicamadas **MLP**, a rede neural nebulosa **ANFIS**, a rede neural nebulosa **RNN**, o modelo periódico autoregressivo **PAR**(p_m) e o modelo evolutivo baseado em regras nebulosas **eTS**.

Os modelos **PAR**(p_m) e **RNN** foram ajustados utilizando doze modelos, um para cada mês. A ordem dos modelos autoregressivos varia de 1 a 3 e o número de regras do modelo **RNN** varia de 6 a 35, dependendo do mês a ser ajustado.

Os resultados de simulação obtidos na previsão um passo à frente, para o período de

Janeiro de 1996 a Dezembro de 2005 são ilustrados na Figura 30. Os resultados de previsão obtidos pelo modelo construtivo são apresentados na Figura 30-(a). A Figura 30-(b) e (c) ilustra as previsões obtidas pelos modelos **A-FSM** e **mA-FSM**, respectivamente.

Tabela 11: Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1981 a 1985.

Model	M^0 initial	M	ERMs (%)	REQM (m^3/s)	EAM (m^3/s)	NDEI
PAR(p_m)	—	(12)	20,93	484,21	317,58	0,5600
MLP	—	—	21,78	469,22	326,21	0,5469
ANFIS	8	8	23,40	445,39	301,36	0,5647
RNN	0	(12)	22,37	466,51	307,83	0,5437
C-FSM	2	24	20,72	448,29	314,04	0,5700
eTS	0	0–6	26,42	541,61	364,14	0,6312
A-FSM	11	2	19,76	484,69	302,42	0,5700
mA-FSM	3	2	19,79	485,16	302,86	0,5700

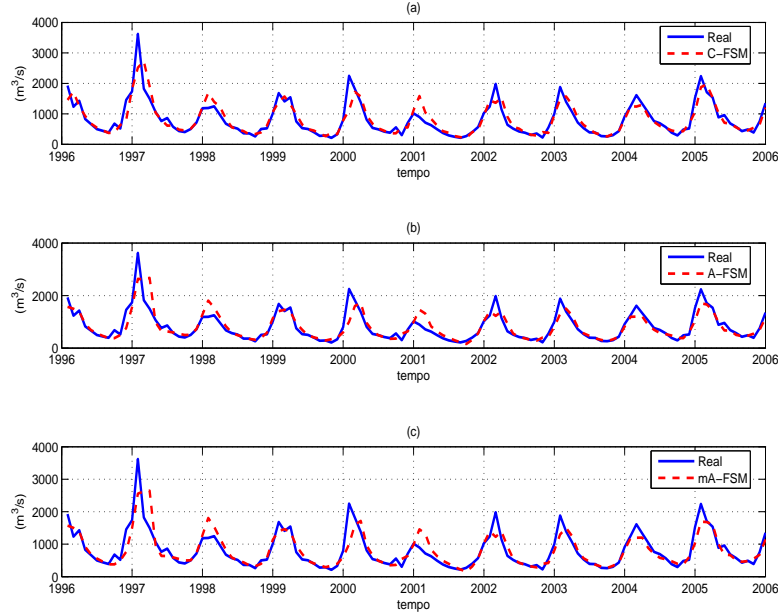


Figura 30: Resultados de previsão um passo à frente para a série de vazões mensais do posto de Furnas: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

Nota-se na Tabela 11, que os modelos propostos apresentaram um bom desempenho, com uma estrutura parcimoniosa.

Os erros globais de previsão para os modelos baseados em regras nebulosas, assim como os resultados obtidos pelos modelos comparativos são apresentados na Tabela 12, para o período de Janeiro de 1996 a Dezembro de 2005. A Tabela 12 também apresenta o número final de regras nebulosas (M) para cada modelo ajustado.

No caso dos modelos não-lineares com aprendizado *offline*, tanto o modelo **MLP** como os modelos **ANFIS** e **RNN** utilizaram o mesmo conjunto de treinamento e de variáveis de entrada utilizado pelo modelo construtivo **C-FSM**. Já no caso do **eTS**, por não ter uma etapa de inicialização do modelo, este utiliza todos os dados de entrada-saída disponíveis para o ajuste sequencial, com as mesmas variáveis de entrada.

Tabela 12: Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, período de 1996 a 2005.

Model	M^0 initial	M	ERMs (%)	REQM (m^3/s)	EAM (m^3/s)	NDEI
PAR(p_m)	–	(12)	17,42	266,13	154,44	0,4772
MLP	–	–	18,12	258,53	161,45	0,4635
ANFIS	8	8	18,85	262,21	166,31	0,4701
RNN	–	(12)	19,10	258,85	169,30	0,4642
C-FSM	2	19	17.58	257.86	158.93	0.4623
eTS	0	0–6	24,96	209,45	209,45	0,5744
A-FSM	9	2–9	18,25	259.05	162.65	0.4700
mA-FSM	3	2	18.50	285.33	170.41	0.5116

Em termos gerais, pode-ser notar que os modelos baseados em regras nebulosas propostos neste trabalho forneceram modelos de série temporal adequados para a série em estudo, com erros de previsão menores que os outros modelos da literatura e validados pela análise de resíduos apresentada. Além disso, os modelos com aprendizado sequencial forneceram resultados de previsão superiores aos obtidos pelo modelo **eTS**, o qual possui uma estratégia de aprendizado similar ao algoritmo sequencial utilizado pelo modelo **A-FSM** e **mA-FSM**. Embora os modelos com aprendizado sequencial tenham apresentado um erro de previsão superior aos obtidos pelos modelos com aprendizado *offline*, deve-se considerar que estes modelos possuem estruturas mais compactas.

Apesar do modelo **eTS** ter apresentado um desempenho inferior nesta aplicação, deve-se

ressaltar que este modelo não possui uma etapa de inicialização, diferente ao modelo **A-FSM**, o que influencia o desempenho do aprendizado sequencial, embora agilize a construção do modelo de previsão.

A Tabela 13 apresenta os erros mensais de previsão para o período de Janeiro de 1996 a Dezembro de 2005, obtidos pelos modelos **C-FSM** e **A-FSM**.

Tabela 13: Erros mensais para os modelos **C-FSM** e **A-FSM**, período de 1996 a 2005.

	C-FSM			A-FSM		
Mês	ERMs	REQM	EAM	ERMs	REQM	EAM
	(%)	(m^3/s)	(m^3/s)	(%)	(m^3/s)	(m^3/s)
Jan	29,69	621,06	515,28	29,68	608,29	515,29
Feb	22,73	429,49	355,69	23,89	444,99	371,59
Mar	17,21	249,47	216,41	17,58	261,72	221,85
Abr	18,20	160,95	147,55	18,94	168,36	152,81
Mai	13,00	108,95	82,70	14,31	112,26	89,41
Jun	13,59	100,00	72,59	13,27	97,87	70,11
Jul	6,59	34,45	28,13	7,684	39,85	30,62
Ago	10,14	41,90	35,56	11,37	44,98	39,28
Set	23,20	124,98	92,00	22,75	123,55	90,62
Out	27,59	133,69	107,88	30,76	140,57	120,62
Nov	19,68	233,51	144,11	19,80	227,69	143,07
Dez	9,30	164,97	109,27	8,93	166,51	106,50

A modo de comparação, os modelos **C-FSM** e **A-FSM** serão ajustados utilizando a abordagem utilizada pelo modelo **PAR**(p_m). Ou seja, um modelo é ajustado para cada mês da série de vazão, com entradas selecionadas utilizando o algoritmo FNN-PMI, para cada um dos doze modelos de previsão.

A Tabela 14 apresenta os resultados obtidos durante a seleção de entradas para cada um dos doze modelos de previsão, assim como a ordem dos modelos periódicos autoregressivos ajustados e o número de regras do modelo de rede neural nebulosa **RNN**. Estas entradas serão utilizadas para o ajuste de doze modelos construtivos **C-FSM**, assim como para o ajuste de doze modelos sequenciais **A-FSM**.

Tabela 14: Seleção de entradas para os modelos construtivos, ordem dos modelos auto-regressivos e número de regras dos modelos **RNN** para a série de vazões mensais de Furnas.

Modelo	Modelos baseados em regras			PAR	RNN
Mês	Entradas	PMI	Limiar (95%)	Ordem (p)	M
Janeiro	1	0,2997	0,2181	1	18
	13	0,1858	0,1606		
	8	0,1240	0,1082		
	11	0,0424	0,0479		
Fevereiro	1	0,2677	0,2293	2	35
	9	0,1708	0,1877		
Março	1	0,3036	0,2236	1	21
	14	0,1898	0,1837		
	3	0,0929	0,0906		
	3	0,0309	0,0351		
Abril	1	0,4405	0,2314	2	35
	2	0,1441	0,1176		
	14	0,1138	0,0962		
	8	0,0359	0,0393		
Maio	1	0,4892	0,2292	3	14
	3	0,1568	0,1067		
	13	0,0770	0,0778		
Junho	1	0,4680	0,1711	1	6
	3	0,0774	0,0561		
	7	0,0433	0,0442		
Julho	2	0,5210	0,2067	1	30
	7	0,0965	0,1037		
Agosto	1	0,5432	0,2187	2	20
	8	0,1205	0,1099		
	12	0,0446	0,0503		
Setembro	1	0,4034	0,1947	1	14
	6	0,0854	0,0716		
	10	0,0757	0,0672		
	4	0,0347	0,0330		
Outubro	2	0,2921	0,1907	3	13
	10	0,1385	0,1159		
	9	0,0825	0,0745		
	14	0,0406	0,0362		
Novembro	1	0,3388	0,1749	1	28
	9	0,1255	0,1224		
	12	0,0936	0,0989		
Dezembro	1	0,3212	0,2232	1	28
	11	0,1352	0,1387		

Embora o modelo com aprendizado sequencial precise de uma seleção de entradas diferenciada, considerando apenas o período do histórico utilizado para a inicialização da estrutura do modelo, neste caso utilizou-se as mesmas entradas utilizadas pelo modelo construtivo, já que o número inicial de anos utilizados para inicializar os modelos é igual a 20, o que equivale a um total de 20 dados disponíveis para efetuar a seleção de entradas de cada modelo mensal, sendo esta quantidade de dados insuficiente para as estimativas do PMI.

Além disso, estes ajustes são apresentados apenas a modo de comparação, para assim verificar a existência de ganhos nos resultados de previsão nesta aplicação em particular, desagregando o problema em problemas menores, independentes e utilizando as abordagens propostas neste trabalho.

O procedimento adotado para o ajuste dos 12 modelos mensais foi o mesmo utilizado no decorrer deste trabalho, para cada um dos modelos ajustados. A Tabela 15 apresenta os erros globais de previsão para o período de validação considerado.

Tabela 15: Erros de previsão um passo à frente para a série de vazões mensais do posto de Furnas, modelagem mês a mês para o período de 1996 a 2005.

Model	ERMs (%)	REQM (m^3/s)	EAM (m^3/s)	NDEI
C-FSM	17,67	279,34	161,79	0,50
A-FSM	16,79	230,75	140,28	0,49

Comparando os resultados da Tabela 12 para os modelos **C-FSM** e **A-FSM**, com os resultados apresentados na Tabela 15, observa-se que apenas o modelo **A-FSM** apresentou um ganho de desempenho de aproximadamente 8% em termos de **ERMs**, quando comparado com o resultado obtido ajustando um único modelo.

Os erros mensais para a previsão um passo à frente dos dez últimos anos do histórico da série são apresentados na Tabela 16.

Tabela 16: Erros globais para os modelos **C-FSM** e **A-FSM**, modelagem mês a mês para o período de 1996 a 2005.

	C-FSM			A-FSM		
Mês	ERMs	REQM	EAM	ERMs	REQM	EAM
	(%)	(m^3/s)	(m^3/s)	(%)	(m^3/s)	(m^3/s)
Jan	29,95	714,51	541,41	22,06	466,55	340,62
Feb	22,37	431,17	344,07	21,91	396,96	309,28
Mar	16,37	232,80	193,36	18,88	270,65	223,08
Abr	19,77	192,32	165,87	16,38	170,56	141,71
Mai	12,93	112,20	82,96	11,33	107,52	73,75
Jun	11,25	92,81	64,34	11,06	90,02	59,24
Jul	8,19	42,52	34,71	7,55	40,38	33,41
Ago	11,25	42,12	39,04	11,41	42,39	38,68
Set	20,61	109,84	82,01	20,30	120,39	85,21
Out	25,06	117,61	94,49	28,50	128,28	104,33
Nov	23,32	253,07	168,07	24,11	269,10	174,78
Dez	11,01	184,64	131,18	8,04	186,80	99,27

Comparando a Tabela 16 com os resultados apresentados na Tabela 13, observa-se que os resultados foram bastante similares, principalmente para o modelo **C-FSM**, sendo que, em alguns meses do período de validação, os erros aumentaram.

Por outro lado, o modelo **A-FSM** obteve algumas melhorias no desempenho adotando esta abordagem de modelagem, o que levou a uma diminuição do erro global para o período de validação considerado, o que verifica o ganho apresentado para este período do histórico da série, como mostra a Tabela 15.

Dos resultados obtidos, pode-se observar que, a maior dificuldade para os modelos propostos nesta aplicação encontra-se na previsão nos períodos de chuva, período em geral de maior variação e complexidade para modelar e prever. Além disso, os resultados mostram que, principalmente no caso do modelo **C-FSM**, a desagregação da modelagem da série de vazões mensais, em doze modelos construtivos de séries temporais não fornecem ganho de desempenho na previsão da série, para o período de 1996 a 2005, sendo assim, a abordagem de modelagem por meio de um único modelo a mais adequada em termos de complexidade e esforço computacional.

7.6 Série de vazões semanais

Nesta seção analisa-se os dados de vazões afluentes semanais da usina de Furnas, os quais compõem uma série histórica que abrange o período a partir da primeira semana de Janeiro de 1931 à última semana de Dezembro de 2000. Neste caso, a análise da série foi feita assumindo a construção de um único modelo de previsão para a série de vazões semanais.

Como o objetivo desta seção consiste tanto em identificar, estimar os parâmetros e realizar a previsão, o conjunto de dados foi dividido em dois subconjuntos: um conjunto de treinamento, usado para a identificação e estimação do modelo; e um conjunto de teste, usado para a previsão. O conjunto de teste corresponde ao período a partir da semana 45 (primeira de Novembro) do ano de 1999 até a semana 17 (última de Abril) do ano 2000. Para tal, o conjunto de treinamento corresponde ao histórico de dados até a semana anterior à previsão (Janeiro de 1931 a Outubro de 1999).

O algoritmo FNN-PMI foi aplicado para a seleção de entradas. No caso do modelo construtivo, utilizou-se o conjunto completo de dados de treinamento. No caso dos modelos com aprendizado sequencial, o conjunto de dados utilizado para a seleção de entradas é composto pelos 20 primeiros anos do histórico da série.

Inicialmente, o algoritmo FNN é aplicado a série de dados para a determinação do conjunto inicial de entradas para o critério PMI. A Figura 31 ilustra o número de falsos vizinhos (FNN) a medida que um maior número de atrasos é considerado para compor o conjunto inicial de possíveis entradas.

A Figura 31-(a) ilustra o número de falsos vizinhos a cada iteração do algoritmo sob o conjunto de treinamento do modelo construtivo. A Figura 31-(a) ilustra o número de falsos vizinhos para cada iteração do algoritmo FNN utilizando o conjunto de inicialização do modelo **A-FSM**.

Observa-se que, para um número de atrasos igual a 7, o número de falsos vizinhos para ambos os modelos é zero ou próximo a zero, sendo portanto, o conjunto inicial para a aplicação do algoritmo PMI composto pelos sete primeiros atrasos da série.

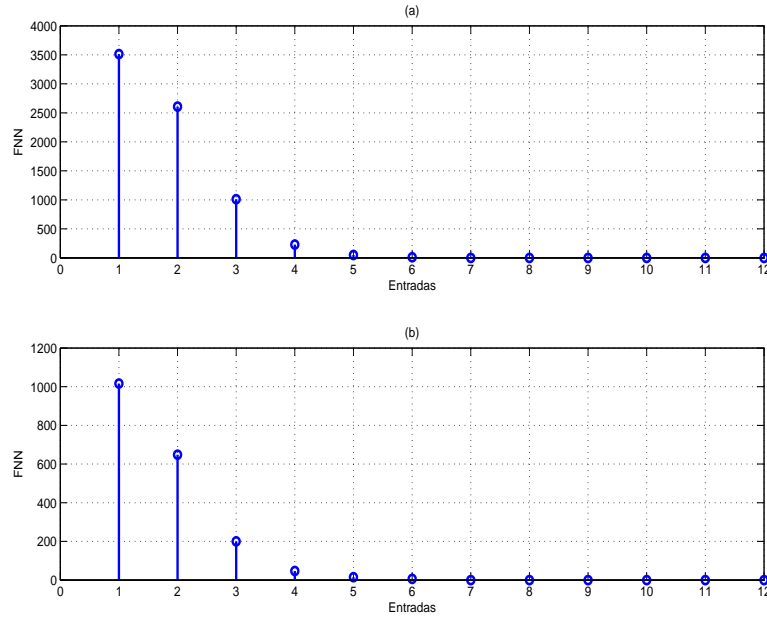


Figura 31: Definição do conjunto inicial de possíveis entradas via o algoritmo FNN, para a série semanal do posto de Furnas: (a) modelo **C-FSM** (68 anos), (b) modelo **A-FSM** (20 anos).

A partir do conjunto inicial de variáveis definido pelo algoritmo FNN, verifica-se o critério PMI para cada possível entrada do modelo a ser construído, definindo um limiar para poda ou finalização do algoritmo igual a 0,05. Desta forma, os atrasos com um valor de PMI inferior a este limiar não serão considerados como variáveis relevantes para a modelagem da série. Assim, os atrasos 1, 7, 4, 2, e 5 são utilizados por ambos os modelos, construtivo e sequencial, como mostra a Tabela 17. Já a contribuição dos atrasos 6 e 3 não é significativa, ou seja, os valores do PMI para esses atrasos são inferiores ao limiar previamente estabelecido.

Após a seleção das entradas, os modelos são ajustados e validados. O limiar para os coeficientes α durante o aprendizado construtivo foi igual a $\alpha_{min} = 0,01$, sendo o processo de aprendizado finalizado após 40 épocas.

O aprendizado sequencial para o modelo **A-FSM** considerou $\alpha_{min} = 0,001$, $T = 52$, $\epsilon = 0,50$ e $f_{forget} = 0,85$. Já o modelo **mA-FSM** inicializou a sua estrutura com três regras nebulosas definidas pelo algoritmo SC, com $r_a = 0,10$ e $r_b = 1,50$. Após a inicialização, o modelo **mA-FSM** foi ajustado sequencialmente adotando $\alpha_{min} = 0,001$, $T = 13$, $\epsilon = 0,50$ e $f_{forget} = 0,85$.

Tabela 17: Resultados aplicando o algoritmo FNN-PMI para a série de vazões semanais do posto de Furnas.

C-FSM(68 anos)			A-FSM (20 anos)		
Lag	PMI	Limiar	Lag	PMI	Limiar
1	0,2849	0,0395	7	0,0893	0,0640
7	0,1539	0,0385	1	0,0858	0,0664
4	0,1219	0,0297	4	0,0709	0,0600
2	0,0873	0,0192	2	0,0670	0,0498
5	0,0554	0,0149	5	0,0545	0,0341
6	0,0431	0,0099	6	0,0465	0,0107
3	0,0353	0,0055	3	0,0352	0,0001

A Figura 32 ilustra a variação do número de regras que compõe cada modelo ajustado.

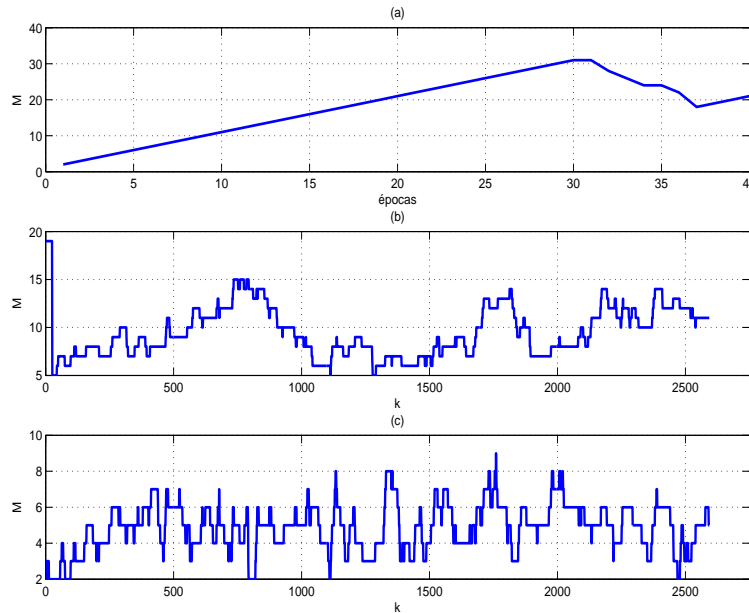


Figura 32: Variação de M para cada um dos modelos baseados em regras ajustados para a série de vazões semanais do posto de Furnas: (a) **C-FSM**, (b) **A-FSM**, (c) **mA-FSM**.

Após o ajuste dos modelos, segue a fase de previsão a partir da primeira semana de Novembro de 1999 até a última semana de Abril de 2000. Este período corresponde ao período de cheia, onde os dados apresentam maior variabilidade.

Os resultados obtidos são comparados aos do modelo PREVIVAZ (modelo estocástico de previsão de vazões semanais), utilizado pelo setor elétrico brasileiro e desenvolvido pelo

Centro de Pesquisas em Energia Elétrica - CEPEL (CEPEL, 1997). O sistema PREVIVAZ consiste no ajuste de 94 modelos com diferentes combinações de séries temporais, estrutura estacionária ou periódica, métodos de estimação de parâmetros e diferentes transformações, selecionando o melhor modelo dentre os 94 para realizar a previsão de vazões semanais.

Ainda para comparação do desempenho dos modelos propostos, foi considerada a rede neural nebulosa (**RNN**). Este modelo foi ajustado no mesmo período que o modelo **C-FSM**, entretanto, para o modelo **RNN**, o número de variáveis de entrada foi igual a 8. Ou seja, para fazer a previsão da semana t , considerou-se como entrada os oito atrasos imediatamente anteriores. O ajuste dos parâmetros da rede neural nebulosa, ou seja, valores modais, dispersão e pesos da rede, foram ajustados para cada semana.

O número de regras variou de 5 a 35 regras, sendo que, para o início do período de chuva utilizado para teste (primeira semana de Novembro de 1999), o número de regras foi menor que para as demais semanas (Ballini et al., 2005).

A análise do desempenho dos modelos de previsão foi avaliada através da raiz do erro quadrático médio (**REQM**), erro absoluto médio (**EAM**) e erro relativo médio (**ERM**(%)), o qual é definido como segue:

$$\mathbf{ERM} = \frac{100}{N} \sum_{k=1}^N \frac{|y^k - \widehat{y}^k|}{y^k} \quad (7.10)$$

sendo N o número de valores previstos, \widehat{y}^k a saída do modelo de série temporal e y^k a saída desejada. Os erros globais de previsão sob o conjunto de validação são apresentados na Tabela 18.

Tabela 18: Erros de previsão um passo à frente para a série de vazões semanais do posto de Furnas.

Model	M^0 initial	M	ERM (%)	REQM (m^3/s)	EAM (m^3/s)
PREVIVAZ	–	(52)	40,90	747,66	471,50
RNN	–	5-35	37,70	701,43	399,50
C-FSM	2	21	32,66	652,78	389,41
A-FSM	19	11	27,11	636,93	382,66
mA-FSM	3	5	31,03	680,68	388,91

Como mostra a Tabela 18, os modelos de série temporal propostos obtiveram um melhor desempenho para o período de validação definido, quando comparados aos modelos PREVIVAZ e **RNN**. O modelo com melhores resultados de previsão foi o modelo **A-FSM**, o qual foi inicializado com $M = 19$ regras nebulosas, sendo a estrutura final do modelo composta por $M = 11$ regras nebulosas, obtendo um **ERM** de 27,11%, o que representa um ganho de 33,6% no **ERM** no desempenho quando comparado ao modelo PREVIVAZ e **RNN**.

Ainda deve ser observado que foi ajustado um único modelo para a previsão da série de vazões semanais, enquanto que o modelo PREVIVAZ seleciona, dentre os 94 modelos ajustados para cada semana, o que apresenta melhor desempenho no conjunto de validação.

7.7 Resumo

Este capítulo apresentou um conjunto de aplicações, visando a geração de modelos adequados e eficientes para previsão de séries temporais. Nesse contexto, utilizou-se a metodologia proposta neste trabalho, inicializando o processo de modelagem por meio da análise das séries, o pré-processamento dos dados, o ajuste dos modelos, para finalizar com a validação dos mesmos via a análise dos resíduos e dos erros na previsão um e vários passos à frente.

Primeiramente, mostrou-se, a modo ilustrativo, os resultados obtidos para a série de Box-Jenkins, conhecida na literatura e utilizada para fins de comparação de desempenho, ajustando os modelos **C-FSM**, **A-FSM** e **mA-FSM**. Os modelos foram comparados aos modelos **eTS** e **simplLeTS**. Os resultados de simulação mostraram a eficiência dos modelos propostos para esta aplicação.

Em seguida, uma série econômica foi analisada e os resultados obtidos aplicando os modelos propostos neste trabalho foram comparados aos resultados obtidos por modelos lineares tradicionais da área de séries temporais, na previsão de um até cinco passos à frente. Neste caso, os modelos propostos neste trabalho forneceram previsões mais acuradas para os diferentes horizontes de previsão. Esta aplicação também mostrou a influência do pré-processamento dos dados no desempenho dos modelos de previsão.

Além disso, três séries financeiras foram modeladas, considerando tanto séries mensais sazonais, como séries não sazonais e com tendência. Os testes apresentados incluem previsões um e até dezoito passos à frente. Os resultados obtidos para estas séries financeiras foram parcialmente submetidos na Competição de Previsão de Séries Temporais NN3'07, obtendo o quinto lugar na avaliação final, para o conjunto reduzido de séries temporais.

Finalmente, a metodologia proposta também foi aplicada na construção de modelos para as séries de vazões mensais e semanais do posto de Furnas.

No caso da série de vazões mensais, os resultados obtidos pelos modelos **C-FSM**, **A-FSM** e **mA-FSM** foram similares ou melhores que os modelos neurais e baseados em regras nebulosas da literatura, para dois períodos de validação diferentes.

A análise dos modelos estendeu-se para uma abordagem desagregada. Ou seja, além do ajuste de um único modelo para a série temporal, realizou-se o ajuste de doze modelos mensais, sendo um modelo ajustado para cada mês. Os resultados de previsão obtidos, para o período de validação considerado, mostraram que o ganho obtido desagregando o problema em doze problemas independentes não justifica o aumento da complexidade da modelagem, quando utilizado o modelo baseado em regras nebulosas **C-FSM**.

No caso do modelo sequencial **A-FSM**, a desagregação resultou em melhor desempenho com um ganho de 8% quando comparado ao modelo **C-FSM**.

Vale ressaltar que os modelos propostos para a aplicação a séries de vazões mensais resultaram em um desempenho similar ou superior às abordagens **PAR**, **MLP**, **ANFIS**, **RNN** e **eTS**, com estruturas parcimoniosas.

Por outro lado, os modelos baseados em regras nebulosas forneceram resultados de previsão de vazões semanais eficientes e melhores quando comparados aos resultados obtidos pelo modelo PREVIVAZ, modelo atualmente utilizado pelo setor elétrico brasileiro, o qual é baseado em modelos tradicionais de série temporais e o modelo **RNN** proposto em (Figueiredo, 1997) e aplicado a séries de vazões semanais em (Ballini et al., 2005).

Os resultados, em geral, mostram a aplicabilidade dos modelos propostos na modelagem de séries temporais de natureza variada, obtendo modelos adequados e eficientes, para diferentes horizontes de previsão.

8 *Conclusões e trabalhos futuros*

8.1 Conclusões

Este trabalho propõe uma metodologia de análise de previsão de séries temporais que leva em conta a capacidade de lidar com relações não lineares não somente a partir da modelagem da série utilizando modelos baseados em regras nebulosas, e sim, a partir de um estágio anterior que é a seleção das variáveis de entradas para o modelo de previsão.

Além disso, a metodologia proposta combina as técnicas estatísticas de análise de séries temporais com ferramentas de modelagem da área de inteligência computacional na construção de modelos não lineares de séries temporais. A metodologia proposta resultou na construção de dois modelos baseados em regras nebulosas. As principais características destes modelos são:

- As estruturas são baseadas em regras nebulosas do tipo Takagi-Sugeno, sendo os seus parâmetros ajustados utilizando procedimentos de aprendizado baseados no algoritmo da Maximização da Verossimilhança (EM), que a partir dos quais surgiram duas propostas de modelagem: a abordagem construtiva offline (**C-FSM**) e a abordagem adaptativa online ou seqüencial (**A-FSM**).
- Os algoritmos de aprendizado propostos tiveram por objetivo principal, além de determinar um conjunto adequado de parâmetros do modelo, determinar automaticamente o número de regras necessárias para compor a estrutura. Isto foi possível gerando operadores de adição de regras nebulosas e de punição de regras, aplicados durante o processo de aprendizado, seja este construtivo ou seqüencial.
- Embora o algoritmo EM seja bastante utilizado para a otimização de modelos baseados em mistura de especialistas e áreas afins, este trabalho estendeu a sua aplicação para sistemas nebulosos, como alternativa ao algoritmo de retropropagação do erro, amplamente utilizado nessa área.

- Ainda que os resultados de previsão sejam, em geral, similares para ambos os modelos **C-FSM** e **A-FSM**, a principal vantagem dos modelos **A-FSM** é o tipo de aprendizado utilizado, já que a partir de um modelo ajustado até um determinado período de tempo, este pode ser re-ajustado considerando apenas a informação mais recente que possa ser fornecida ao modelo. Esta característica torna-se interessante principalmente para problemas de grande porte, que requeiram procedimentos de modelagem e otimização rápidos e eficientes, assim como de estruturas compactas com capacidade de adaptação a mudanças na dinâmica do sistema em estudo.

Os estudos de caso apresentados no Capítulo 7 consideraram séries de comparação, séries econômicas e financeiras, assim como séries de vazões mensais e semanais. Os resultados mostraram os diferentes aspectos da metodologia descritos sucintamente no Capítulo 1 e detalhados no decorrer dos Capítulos 2 a 6. Os casos discutidos mostraram também a influência dos cenários apresentados, incluindo a natureza da série (como por exemplo, a existência de componentes de tendência e/ou sazonalidade), o tamanho do histórico da série, a estrutura do modelo e o algoritmo de aprendizado.

Apesar dos resultados descritos mostrarem, em termos gerais, a proposta como uma alternativa promissora, como foi mencionado no Capítulo 1, este trabalho não pretende fornecer uma metodologia universal de modelagem de séries temporais, e sim uma proposta alternativa que considere os diferentes aspectos envolvidos na análise de séries temporais, e não somente a construção do modelo em si, já que os resultados de previsão assim como o desempenho de quaisquer modelos dependem fortemente de uma série de fatores tais como:

- A natureza dos dados e do problema;
- A quantidade de informação disponível;
- Os critérios de seleção de variáveis utilizados;
- A natureza do modelo (linear ou não-linear);
- O tipo de algoritmo de otimização empregado;
- O período e horizonte de previsão.

Assim, ainda que tenha-se apresentado um conjunto de séries temporais com dinâmicas e características diferentes, assim como cenários de previsão variados e considerando os fatores mencionados, os resultados têm como propósito mostrar a aplicabilidade da metodologia

proposta, assim como diretrizes gerais para a construção de modelos de previsão de forma consistente.

Ainda que os algoritmos de aprendizado propostos e adotados como parte da metodologia descrita neste trabalho forneceram resultados satisfatórios, estes possuem características e aspectos que podem ser melhorados, como parte do desenvolvimento de pesquisas futuras.

8.2 Trabalhos futuros

Os resultados apresentados sugerem que os modelos baseados em regras nebulosas são uma boa alternativa de modelagem de séries temporais, com estrutura simples, embora os algoritmos de aprendizado empregados possam ser considerados sofisticados e complexos. Porém, como foi visto no Capítulo 7, a propagação do erro na previsão múltiplos passos à frente degrada consideravelmente o desempenho dos modelos ajustados. Este resultado era de se esperar, já que tanto o algoritmo de aprendizado construtivo como seqüencial considera apenas os erros na previsão um passo à frente, o que não garante um desempenho adequado para outros horizontes de previsão.

Por tal motivo, é necessário modificar a função objetivo utilizada de tal forma que os erros de previsão múltiplos passos à frente sejam considerados e assim, melhorar o desempenho dos modelos de série temporal no problema de previsão vários passos à frente.

A modificação do algoritmo de seleção de entradas, de tal forma que a seleção realizada para um período da série temporal possa ser re-avaliada em paralelo ao aprendizado seqüencial, já que podem surgir variações na dinâmica da série que não são “explicadas” pelas variáveis selecionadas pelo período anterior. Para realizar estas modificações, é necessário re-escrever as equações utilizadas neste trabalho, o que sugere a busca por estimativas simplificadas para os critérios utilizados, assim como considerações adicionais na otimização seqüencial dos modelos de série temporal.

Outros aspectos que devem ser considerados como propostas futuras são:

- A simplificação dos modelos propostos, em termos de número de parâmetros por regra nebulosa, diminuindo a complexidade do modelo, o qual também influencia o processo de aprendizado.
- A otimização dos operadores de geração de novas regras nebulosas, assim como o critério de poda, de tal forma que o algoritmo seja menos sensível a dados que possam ser

considerados como “*outliers*”, e que, ao mesmo tempo, sejam capazes de lidar com altas variabilidades na série, como é o caso das séries de vazões mensais e semanais.

- Ainda que a metodologia desenvolvida neste trabalho tenha sido aplicada na análise e modelagem de séries temporais, o mesmo procedimento de modelagem pode ser estendida a outras áreas da engenharia, como por exemplo, o controle de processos físicos.

APÊNDICE A – Trabalhos publicados pela autora

1. Luna I., Ballini R., Soares S.; Modelo Adaptativo Baseado em Regras Nebulosas Aplicado à Previsão de Séries de Vazões Semanais; Congresso Brasileiro de Redes Neurais - CBRN'07, Florianópolis, Brasil, Outubro 2007.
2. Luna I., Ballini R., Soares S.; Modelo Construtivo Baseado em Regras para Previsão de Séries Temporais; Simpósio Brasileiro de Automação Inteligente - SBAI'07, Florianópolis, Brasil, Outubro 2007.
3. Luna I., Ballini R., Soares S.; A Constructive-Fuzzy Rule Based System for Time Series Forecasting; International Joint Conference on Neural Networks - IJCNN'07, Orlando, Florida - USA, Agosto 2007, pp. 2908–2913.
4. Luna I., Ballini R., Soares S.; An Adaptive Hybrid Model for Monthly Streamflow Forecasting; IEEE International Conference on Fuzzy Systems - FUZZ-IEEE'07, Londres - UK, Julho 2007, pp. 1–6.
5. Luna I., Ballini R., Soares S.; A Constructive vs. an Online Approach for Time Series Prediction; North American Fuzzy Information Processing Society Meeting - NA-FIPS'07, San Diego, California - USA, Junho 2007, pp. 256–261.
6. Luna I., Ballini R., Soares S.; Técnica de Identificação de Modelos Lineares e Não Lineares de Séries Temporais; Controle e Automação, Setembro 2006, vol. 17, **3**, pp. 245–256.
7. Luna I., Ballini R., Soares S.; Técnica de Identificação de Modelos de Séries Temporais; XVI Congresso Brasileiro de Automática - CBA'06, Salvador - Brasil, Outubro 2006, pp. 1902–1907.

8. Luna I., Soares S., Magalhães M. H. e Ballini R.; Partial Mutual Information Criterion For Modelling Time Series Via Neural Networks, The 11th Information Processing and Management of Uncertainty International Conference, IPMU 2006, Paris-França, Julho 2006, pp. 2012–2019.
9. Luna I., Soares S., Magalhães M. H. e Ballini R.; Previsão de Vazões utilizando Redes Neurais Recorrentes; Congresso Brasileiro de Redes Neurais - CBRN'05, Natal, Outubro 2005.
10. Luna I., Soares S., Magalhães M. H. e Ballini R.; Streamflow Forecasting using Neural Networks and Fuzzy Clustering Techniques; Proceedings of the International Joint Conference on Neural Networks - IJCNN'05, Montreal, Julho 2005, pp. 2631–2636.

Referências

- Abarbanel, H. D. I. e Kennel, M. B. (1993). Local false nearest neighbors and dynamical dimensions from observed chaotic data, *Physical Review E* **47**: 3057–3068.
- Aguirre, L. A. (2004). *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*, 2 edn, Editora UFMG, Belo Horizonte, BH.
- Akaho, S. (2002). Conditionally independent component analysis for supervised feature extraction, *Neurocomputing* **49**: 139–150.
- Akaike, H. (1969). Fitting autoregressive models for prediction, *Annals of Institute of Statistical Mathematics* (21): 243–247.
- Angelov, P. e Filev, D. (2002). Flexible models with evolving structure, *International Journal of Intelligent Systems* **19**(4): 327–340.
- Angelov, P. e Filev, D. (2005). SimplLeTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy models, *Proceedings of The 2005 IEEE International Conference on Fuzzy Systems*, Reno - Nevada, USA, pp. 1068–1073.
- Angelov, P. P. e Filev, D. P. (2004). An Approach to Online Identification of Takagi-Sugeno Fuzzy Models, *IEEE Transactions on Systems, Man and Cybernetics-part B* **34**(1): 484–498.
- Angelov, P. e Zhou, X. (2006). Online Identification of Takagi-Sugeno Fuzzy Models - DEMO. Acesso 8 de Setembro de 2007.
*<http://194.80.34.45/eTS/>
- Ballini, R. (2000). *Análise e Previsão de Vazões Utilizando Modelos de Série Temporais, Redes Neurais e Redes Neurais Nebulosas*, Tese de Doutorado, FEEC-UNICAMP, Brasil.
- Ballini, R., Soares, S., Guilhon, L. G. F. e Gomide, F. (2005). Previsão de Vazões Semanais Utilizando Redes Neurais Nebulosas. Congresso Brasileiro de Redes Neurais - CBRN'05.
- Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, USA: Plenum Press.
- Bonnlander, B. V. e Weigend, A. S. (1994). Selecting input variables using mutual information and nonparametric density estimation, *Proceedings of the 1994 Int. Symp. on Artificial Neural Networks (ISANN'94)*, Tainan, Taiwan, pp. 42–50.

- Bowden, G. J. (2003). *Forecasting Water Resources Variables Using Artificial Neural Networks*, Tese de Doutorado, University of Adelaide, Australia.
- Bowden, G. J., Maier, H. R. e Dandy, G. C. (2005). Input determination for neural network models in water resources applications. Part 1—background and methodology, *Journal of Hydrology* (301): 75–92.
- Box, G. E. P., Jenkins, G. M. e Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*, 3th edn, Holden Day, Oakland, California, USA.
- Brockwell, P. J. e Davis, R. A. (2002). *Introduction to Time Series and Forecasting*, 2nd edn, Springer, New York, USA.
- Caminhas, W., Tavares, H., Gomide, F. e Pedrycz, W. (1999). Fuzzy sets based neural networks: Structure, learning and applications, *Journal of Advanced Computational Intelligence* **3**(3): 151–157.
- Carvalho, A. X. e Tanner, M. A. (2005). Mixture-of-Experts of Autoregressive Time Series: Asymptotic Normality and Model Specification, *IEEE Transactions on Neural Networks* **16**(1): 39–56.
- CEPEL (1997). *Modelo de Previsão de Vazões Semanais Aplicado ao Sistema Hidroelétrico Brasileiro - PREVIVAZ*. Relatório Técnico, número 125/97.
- Chiu, S. (1994). A cluster estimation method with extension to fuzzy model identification, *Proceedings of the Third IEEE Conference on Fuzzy Systems*, Vol. 2, Orlando - Florida, USA, pp. 1240–1245.
- Cicogna, M. (2003). *Sistema de Suporte à Decisão para o Planejamento e a Programação da Operação de Sistemas de Energia Elétrica*, Tese de doutorado, FEEC-UNICAMP, Brasil.
- Cover, T. e Thomas, J. (1991). *Elements of Information Theory*, 1 edn, John Wiley & Sons, New Jersey, USA.
- de Holanda Sales, P. R., de Bragança Pereira, B. e de Mesquita Vieira, A. (1998). Procedimentos Lineares para Identificação e Estimação dos Parâmetros de Modelos, *Revista Brasileira de Estatística* **59**(212): 25–51.
- Dempster, A., Laird, N. e Rubin, D. (1977). Maximum likelihood for incomplete data via the EM algorithm, *Journal of the Royal Statistical Society B* **39**: 1–38.
- Er, M. J. e Wu, S. (2002). A fast learning algorithm for parsimonious fuzzy neural systems, *Fuzzy Sets and Systems* **126**: 337–351.
- Figueiredo, M. (1997). *Redes neurais nebulosas aplicadas em problemas de modelagem e controle autônomo*, Tese de doutorado, FEEC - UNICAMP, Brasil.

- Fraser, A. M. e Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information, *Physical Review A* **33**(2): 1134–1140.
- Fuzzy Logic Toolbox, For Use with MATLAB®* (2000). 2 edn, The Math Works, Inc., Natick, MA.
- Guo, P. e Xu, L. (1999). Relationship between Mixture of Experts and Ensemble Neural Networks, *Proceedings of the 6th International Conference on Neural Information Processing* **1**: 246–250.
- Guyon, I. e Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**: 1157–1182.
- Haber, R. e Unbehauen, H. (1990). Structure identification of nonlinear dynamic systems – a survey on input/output approaches, *Automatica* **26**: 651–677.
- Hamilton, J. D. (1994). *Time Series Analysis*, Princeton University Press, New Jersey, USA.
- Harris, C. J. e Hong, X. (2001). Neurofuzzy mixture of experts network parallel learning and model construction algorithms, *IEE Proceedings on Control Theory and Application* **148**(6): 456–465.
- Haykin, S. (1994). *Neural Networks, A Comprehensive Foundation*, Macmillan Publishing Company.
- Haykin, S. (2001a). *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc. , New York, USA.
- Haykin, S. (2001b). *Redes Neurais, Princípios e Prática*, 2 edn, Bookman.
- Hong, X. e Harris, C. J. (2002). A mixture of experts network structure construction algorithm for modelling and control, *Applied Intelligence* **16**(1): 59–69.
- Hush, D. e Horne, B. (1993). Progress in Supervised Neural Networks, *IEEE Signal Processing Magazine* **10**(1): 8–39.
- Iyoda, E. (2000). *Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas*, Tese de Mestrado, FEEC-UNICAMP, Brasil.
- Jacobs, R., Jordan, M., Nowlan, S. e Hinton, G. (1991). Adaptive Mixture of Local Experts, *Neural Computation* **3**(1): 79–87.
- Jang, J.-S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Transactions on Systems, Man and Cybernetics* **23**(3): 665–685.
- Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine, *Proceedings of the 8th Annual Conference of the Cognitive Science Society* pp. 531–546.

- Jordan, M. e Jacobs, R. (1993). Hierarchical Mixtures of Experts and the EM Algorithm, *Technical Report AIM-1440*.
*citeseer.ist.psu.edu/jordan93hierarchical.html
- Kasabov, N. K. e Song, Q. (2002). DENFIS: Dynamic Evolving Neural-Fuzzy Inference Systems and Its Application for Time-Series Prediction, *IEEE Transactions on Fuzzy Systems* **10**(2): 144–154.
- Kihoro, J. M., Otieno, R. O. e Wafula, C. (2004). Seasonal time series forecasting: a comparative study of ARIMA and ANN models, *African Journal of Science and Technology* **5**(2): 41–49.
- Kohavi, R. e John, G. H. (1997). Wrappers for feature selection, *Artificial Intelligence* **97**: 273–324.
- Lázaro, M., Santamaría, I. e Pantaleón, C. (2003). A new EM-based training algorithm for RBF networks, *Neural Networks* **16**(1): 69–77.
- Lee, C. e Teng, C. (2000). Identification and control of dynamic systems using recurrent fuzzy neural networks, *IEEE Transactions on Fuzzy Systems* **4**(8): 349–366.
- Lee, M., Lee, S.-Y. e Park, C. (1995). Neurofuzzy controller design using neurofuzzy identifier, *International Journal of Approximate Reasoning* **13**: 269–285.
- Leng, G., McGinnity, T. e Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, *Fuzzy Sets and Systems* **150**(2): 211–243.
- Ljung, L. (1999). *System Identification: Theory for the User*, Prentice-Hall, Upper Saddle River, NJ.
- Luna, I. (2003). *Redes Neurais Fuzzy Aplicadas em Identificação e Controle de Sistemas*, Tese de Mestrado, FEEC-UNICAMP, Brasil.
- Luna, I., Ballini, R. e Gomide, F. (2003). Sistema Híbrido Recorrente para Modelaje y Control de Sistemas Dinámicos no Lineales, *I Congreso Internacional de Científicos Peruanos, Lima - Perú*.
- Luna, I., Soares, S. e Ballini, R. (2006a). Partial Mutual Information Criterion For Modelling Time Series Via Neural Networks, *Proceedings of the 11th Information Processing and Management of Uncertainty International Conference*, Vol. 1, pp. 2012–2019.
- Luna, I., Soares, S. e Ballini, R. (2006b). Técnica de Identificação de Modelos Lineares e Não Lineares de Séries Temporais, *Controle e Automação* **17**(3): 245–256.
- Luna, I., Soares, S., Magalhães, M. H. e Ballini, R. (2005). Streamflow Forecasting using Neural Networks and Fuzzy Clustering Techniques, *Proceedings of the International Joint Conference on Neural Networks - IJCNN'05*.

- Ma, S., Ji, C. e Farmer, J. (1997). An efficient EM-based training algorithm for feedforward neural networks, *Neural Networks* **10**(2): 243–256.
- Makridakis, S. e Hibon, M. (2000). The M-3 Competition: results, conclusions and implications, *International Journal of Forecasting* **16**: 541–476.
- Mao, J., Zhag, J., Yue, Y. e Ding, H. (2005). Adaptive-Tree-Structre-based Fuzzy Inference System, *IEEE Transctions on Fuzzy Systems* **13**(1): 1–12.
- Martins, D. C. (2004). *Redução de dimensionalidade utilizando entropia condicional média aplicada a problemas de bioinformática e de processamento de imagens*, Dissertação de Mestrado, IME-USP, Brasil.
- McLachlan, G. e Basford, K. (1988). *Mixture models: inference and applications to clustering*, Marcel Dekker, Inc. SERIES: statistics: textbooks and monographs volume 84.
- Mitra, S. e Hayashi, Y. (2000). Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework, *IEEE Transactions on Neural Networks* **11**(3): 748–768. May.
- Moerland, P. (1997). Some Methods for Training Mixtures of Experts. Relatório Técnico, IDIAP-Com 97-05.
*<http://citeseer.ist.psu.edu/40536.html>
- Montgomery, A. L., Zarnowitz, V., Tsay, R. S. e Tiao, G. C. (1998). Forecasting the U.S. Unemployment Rate, *Journal of the American Statistical Association* **93**(442): 478–493.
- Moon, Y.-I., Rajagopalan, B. e Lall, U. (1995). Estimation of mutual information using kernel density estimators, *Physical Review E* **52**(3): 2318–2321.
- Morettin, P. A. e Toloi, C. M. C. (2004). *Análise de Séries Temporais*, 1 edn, Edgard Blücher Ltda.
- Murray-Smith, R. e Johansen, T. A. (1995). *Local learning in local model networks*.
- Murray-Smith, R. e Johansen, T. A. (1997). *Multiple Model Approaches to Modelling and Control*, Taylor & Francis.
- Nie, J. (1997). Nonlinear time-series forecasting: A fuzzy-neural approach, *Neurocomputing* **16**: 63–76.
- Pedrycz, W. e Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, Cambridge, MA.
- Quantitative Micro Software (1995). Econometric Views 2.0.
- Ramamurti, V. e Ghosh, J. (1996). Structural Adaptation in Mixture of Experts, *13th International Conference on Pattern Recognition* **4**: 704–708.

- Reyhani, N., Hao, J., Ji, Y. e Lendasse, A. (2005). Mutual information and gamma test for input selection, *Proceedings of the European Symposium on Artificial Neural Networks, ESANN 2005*, Bruges, Belgium, pp. 503–508.
- Rhodes, C. e Morari, M. (1997). False-nearest-neighbors algorithm and noise-corrupted time series, *Physical Review E* **55**(5): 6162–6170.
- Rong, H.-J., Sundararajan, N., Huang, G.-B. e Saratchandran, P. (2006). Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction, *Fuzzy Sets and Systems* (157): 1260–1275.
- Sato, M. (2000). On-line EM Algorithm for the Normalized Gaussian Network, *Neural Computation* **12**: 407–432.
- Sato, M. e Ishii, S. (1999). Reinforcement learning based on on-line EM algorithm, *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems*, Vol. 11, MIT Press, in press, pp. 1052–1058.
- Schwarz, G. (1978). Estimating the dimension of a model, *Annals of Statistics* **6**(2): 461–464.
- Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice and Visualization*, 1 edn, John Wiley & Sons Inc.
- Sharma, A. (2000). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 – A strategy for system predictor identification, *Journal of Hydrology* (239): 232–239.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, 1 edn, Chapman and Hall.
- Solomatine, D. e Siek, M. (2006). Modular learning models in forecasting natural phenomena, *Neural Networks* **10**(2): 215–224.
- Soofi, E. S. (2000). Principal information theoretic approaches, *Journal of the American Statistical Association* pp. 1349–1353.
- Specht, D. F. (1991). A General Regression Neural Network, *IEEE Transactions on Neural Networks* **2**(6): 568–576.
- Takagi, T. e Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man and Cybernetics* **15**(1): 116–132.
- Takens, F. (1981). Detecting strange attractors in turbulence, *Proceedings of Dynamical Systems and Turbulence - Lecture Notes in Mathematics* **898**: 366–381.
- Tsay, R. S. (2005). *Analysis of Financial Time Series*, 2nd edn, Wiley & Sons.
- Wang, L. (1994). *Adaptive Fuzzy Systems and Control*, Prentice Hall.

- Warner, B. e Misra, M. (1996). Understanding Neural Networks as Statistical Tools, *The American Statistician* **50**(4): 284–293.
- Yager, R. e Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*, John Wiley and Sons. Inc.
- Zaffalon, M. e Hutter, M. (2002). Robust features selection by mutual information distributions, *18th International Conference on Uncertainty in Artificial Intelligence*, pp. 577–584.
- Zealand, C. M., Burn, D. H. e Simonovic, S. P. (1999). Short term streamflow forecasting using artificial neural networks, *Journal of Hydrology* (214): 32–48.
- Zheng, G. L. e Billings, S. A. (1995). Radial Basis Function Network Configuration Using Mutual Information and the Orthogonal Least Squares Algorithm, *Neural Networks* **9**(9): 1619–1637.